Department of Computer Science
Series of Publications A
Report A-2019-11

Advances in Motion Sensing on Mobile Devices

Samuli Hemminki

Doctoral dissertation, to be presented for public examination with the permission of the Faculty of Computer Science of the University of Helsinki, in Athena, Lecture Hall 302, on the 25th of November, 2019 at 12 o'clock.

University of Helsinki
Finland

Supervisor
Petteri Nurmi, University of Helsinki, Finland
Sasu Tarkoma, University of Helsinki, Finland

Pre-examiners
Susanna Pirttikangas, University of Oulu, Finland
Daniel Roggen, University of Sussex, UK

Opponent
Moustafa Youssef, Alexandria University, Egypt

Custos
Petteri Nurmi, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Pietari Kalmin katu 5)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://cs.helsinki.fi/
Telephone: +358 2941 911

ISSN 1238-8645
ISBN 978-951-51-5597-9 (paperback)
ISBN 978-951-51-5598-6 (PDF)
Computing Reviews (1998) Classification: C.4, I.5.2, I.5.4, H.4.m
Helsinki 2019
Unigrafia

Advances in Motion Sensing on Mobile Devices

Samuli Hemminki

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
samuli.hemminki@cs.helsinki.fi

PhD Thesis, Series of Publications A, Report A-2019-11
Helsinki, November 2019, 113 + 89 pages
ISSN 1238-8645
ISBN 978-951-51-5597-9 (paperback)
ISBN 978-951-51-5598-6 (PDF)

Abstract

Motion sensing is one of the most important sensing capabilities of mobile devices, enabling monitoring physical movement of the device and associating the observed motion with predefined activities and physical phenomena. The present thesis is divided into three parts covering different facets of motion sensing techniques. In the first part of this thesis, we present techniques to identify the gravity component within three-dimensional accelerometer measurements. Our technique is particularly effective in the presence of sustained linear acceleration events. Using the estimated gravity component, we also demonstrate how the sensor measurements can be transformed into descriptive motion representations, able to convey information about sustained linear accelerations. To quantify sustained linear acceleration, we propose a set of novel peak features, designed to characterize movement during mechanized transportation. Using the gravity estimation technique and peak features, we proceed to present an accelerometer-based transportation mode detection system able to distinguish between fine-grained automotive modalities. In the second part of the thesis, we present a novel sensor-assisted method, crowd replication, for quantifying usage of a public space. As a key technical contribution within crowd replication, we describe construction and use of pedestrian motion models to accurately track detailed motion information. Fusing the pedestrian models with a positioning system and annotations about visual observations, we generate enriched trajectories able to accurately quantify usage of public spaces. Finally in the third part of the thesis, we present two exemplary mobile applications leveraging motion information. As the first application, we present

a persuasive mobile application that uses transportation mode detection to promote sustainable transportation habits. The second application is a collaborative speech monitoring system, where motion information is used to monitor changes in physical configuration of the participating devices.

Computing Reviews (1998) Categories and Subject Descriptors:
C.4 [Computer Systems Organization] Performance of Systems: Modeling techniques
I.5.2 [Pattern Recognition] Design Methodology: Feature evaluation and selection
I.5.4 [Pattern Recognition] Applications: Signal processing
H.4.m [Information Systems]: Information Systems Applications: Miscellaneous

General Terms:
Algorithms, Experimentation, Measurement

Additional Key Words and Phrases:
Mobile Sensing, Motion Sensing, Transportation Mode Detection, Urban Computing, Crowd Replication, Collaborative Sensing

Acknowledgements

I would like to extend my appreciation and gratitude to my supervisors Petteri Nurmi and Sasu Tarkoma. Your feedback, advice and support has been invaluable throughout the years of our collaboration and research. Special thanks to Petteri Nurmi and Patrik Floréen for the nurturing early years within the Adaptive Computing group; I have never quite felt as home within the academic life before or since our adventures in B333. I would also like to extend special thanks to Petri Myllymäki for being the one to guide me into research and the academic world. I would not be here today without your help.

I would like to sincerely thank my opponent Moustafa Youssef for both his enduring contributions in the field, and making the time to accept our invitation to participate as the thesis opponent. I have also greatly benefited from the excellent reviews from both pre-examiners, Susanna Pirttikangas and Daniel Roggens, and their encouraging and helpful comments on the thesis.

I gratefully acknowledge the generous funding provided by Doctoral Program in Computer Science (DoCS), Nokia Foundation Scholarship reward, Academy of Finland, TEKES, ICT program of the European Community ICT-216886-PASCAL2, DIGILE, the Department of Computer Science, and the University of Helsinki.

I am grateful for the advice and companionship of my former and current colleagues both in academia and industry. In particular, I would like to thank Teemu Pulkkinen, Sourav Bhattacharya, Joel Pyykkö, Taneli Vähäkangas, Andreas Forsblom, Yiyun Shen, Jara Uitto, Yina Ye, Tony Kovanen, Haipeng Guo, and Joonas Kukkonen from the Adaptive Computing group. I would also like to extend my appreciation to former colleagues Antti Jylhä, Miika Sirén, Kai Zhao, Aaron Yi Ding, Martti Rannanjärvi, Muneeba Raja, Anja Bachmann, and Stephen Sigg.

A very special thanks to the excellent research group of the Konomi Labs at the CSIS department of University of Tokyo, Shin'ichi Konomi, Keisuke Kuribayashi, Tomoyo Sasao, and Kazuki Wakasa. The four months

in Japan were among the best winters for both me and our family's lives. I warmly thank people I have had the privilege of working with outside academic research, in particular the gentlemen at UK, Andreas Zacharias, Peter Lindgren, Michalis Vitos, and Telmo Amaro. Thank you for expanding my horizons with everything that real-world implementations entail.

I would like to take the opportunity to extend my gratitude to the people supporting and helping during the writing process. The friendly staff at Lammi research station who were kind enough to let me visit their place for a writing retreat, and Riitta Alander for stepping up and helping our family when we were running out of hours and hands.

Finally, and most importantly, I am forever grateful to my family, Marja Alander, Helmi Hemminki and Urho Hemminki, Päivi Saxlund, Christer Kämpe, Kaarina Holmqvist, for your help, patience, support and inspiration. You are what makes it all worthwhile.

Helsinki, November 1st, 2019
Samuli Hemminki

Contents

Original publications

The thesis is based on the following list of original publications, which are referred to as Articles I – VI. The articles are reprinted at the appendix of this thesis.

Article I Samuli Hemminki, Petteri Nurmi, Sasu Tarkoma.

Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th International Conference on Embedded Networked Sensor Systems (SenSys'13, Rome, Italy, November 2013).* ACM, 2013.

Article II Samuli Hemminki, Petteri Nurmi, Sasu Tarkoma.

Gravity and linear acceleration estimation on mobile devices. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous'14, London, UK, December 2014).*

ACM, 2014.

Article III Silvia Gabrielli, Paula Forbes, Antti Jylhä, Simon Wells, Mika Sirén, Samuli Hemminki, Petteri Nurmi, Rosa Maimone, Judith Masthoff, Giulio Jacucci.

Design challenges in motivating change for sustainable urban mobility. In *Computers in Human Behavior, 41 (2014), pp. 416-423.* Elsevier, 2014.

Article IV Samuli Hemminki, Keisuke Kuribayashi, Shin'ichi Konomi, Petteri Nurmi, Sasu Tarkoma.

Quantitative evaluation of public spaces using crowd replication. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPACIAL '16, California, USA, October-November 2016.* ACM, 2016.

Chapter 1

Introduction

The modern world is well characterized by *movement*. Through continuing urbanization, people are increasingly moving into cities, with estimated 2.5 billion more people living in urban areas by 2050 [99]. The expanding cities themselves are constantly in motion with people and goods moving between destinations. The efficiency of a city is thus dependent on the quality of its transportation network, which provides the means for people and goods to reach their destinations and goals. Likewise, the modern global trade economy is strongly based on efficient movement of goods and assets, ranging from logistics within cities to freight transportation between continents. Besides benefits, transportation also brings challenges. For example, in 2014 transportation was responsible for 24% of the world's CO2-emissions [119], and 30% of total EU energy usage [146], stressing the importance of efficient and eco-friendly transportation. Besides the necessary movement to transfer between destinations, people also move to maintain health and well-being, for relaxation, and for social activities. Supporting these activities through urban planning and well-designed public spaces is a vital factor for psychological well-being of a city and its inhabitants [41].

Mobile devices have emerged as one of the main instruments for tracking movement of people through the use of device-embedded motion sensors. Mobile devices are a particularly fitting platform for tracking movement due to their being naturally carried by the users. Being able to capture detailed information about movement allows researchers and developers to extract useful information for tracking, measuring and classifying objects and people. Specifically, motion information empowers both tracking physical properties, e.g., acceleration, velocity and heading of the device [64], as well as recognizing physical activities and phenomena [71]. Such applications of motion information have been shown to have a high impact across several domains, such as transportation [96, 116], health and well-being [89],

security [59], and sports [5]. Transportation in particular can benefit from passively sensed motion information, as the very nature of transportation activities is to move between destinations. The physical properties of motion information, detailed in Section 2.4, can assist in tracking the relative position of both pedestrians [113], and transportation vehicles [150]. Machine learning-based motion inference, detailed in Section 2.5, on the other hand, can be used to associate motion information with transportation modes [52] and specific routes [98]. More careful analysis of motion information can be used to uncover details about movement, such as pace and intensity of pedestrian movement [51], or identify driving style [58] and fuel-efficiency [42] within mechanized transportation.

Contemporary mobile devices, such as smartphones and watches, come equipped with one or more motion sensors, i.e., accelerometers, gyroscopes and magnetometers, enabling vast potential for tracking motion information. However, mobile devices enforce a challenging operating environment for motion sensing systems. The main challenges are related to variation in availability and quality of sensors [128], and unconstrained device placement and orientation [83]. Hardware variation is due to the plethora of commercial smart devices available for users, while unconstrained placement and orientation are due to the mobile device being placed in an arbitrary place, such as trouser pocket, bag or held in hand. From the perspective of a motion sensing system, the unknown device orientation results in a dynamic misalignment between local and global coordinate systems. The ensuing inability to directly measure motion information respective to the global reference frame is particularly challenging for physical motion sensing systems [113, 153]. Machine learning-based motion inference systems are likewise compromised by unconstrained device placement, as human bio-mechanics inherently result in different motion patterns for different sensor placements [24, 108]. Unknown and dynamic device orientation additionally requires transforming sensor measurements into a motion representation which is robust against orientation changes, either by resolving required transformation to a common reference frame [150], or by using an orientation-agnostic motion representation, such as the L^2-norm monitoring the overall force magnitude of the sensor [116]. However, using such orientation-agnostic motion representation comes at a cost of losing the three-dimensional decomposition of motion, thus forfeiting directional information of the motion. Additionally, L^2-norm is naturally dominated by the force of gravity, which can mask even notable sustained linear accelerations on the horizontal plane. The combination of these limitations is prohibitive for physical motion sensing systems tracking heading and

velocity, as well as particularly problematic for applications dependent on measuring movement on the horizontal plane.

Transforming motion information into a global reference frame is able to overcome the main limitations of using orientation-agnostic motion representation, such as the L^2-norm, by retaining the internal geometry of the motion sensor measurements. The first step for the reference frame transformation is typically estimation of the gravity component within the accelerometer measurements. Existing solutions for gravity component estimation generally perform relatively well during physical human movement [84], and when highly accurate sensor hardware is available [64]. However, they break during sustained linear accelerations commonly experienced within mechanized transportation. Additionally, heterogeneous sensing hardware and operating software environments [128], combined with unconstrained device placement and frequent device interaction by the user [101], can produce situations where motion sensing systems struggle to generalize.

The present thesis is divided into three parts, with each part considering a different aspect of motion sensing on mobile devices. In the first part, Chapter 2 first provides background information for topics relevant to this thesis, followed by Chapter 3 detailing our contributions for reference frame transformation, estimation of linear acceleration, and accelerometer-based transportation mode detection. In the second part of this thesis, covering Chapter 4, we shift the focus to pedestrian activities, and present a method coined *crowd replication* to quantify usage of public spaces based on detected pedestrian activities. At the end of the first two parts, we summarize contributions presented in the relevant chapters, and provide a list of additional evaluations presented within the related articles. In the third part of this thesis, covering Chapter 5, we present two example mobile applications leveraging motion sensing for their key functionalities. Finally, in Chapter 6, we discuss limitations of the presented contributions and identify directions for future research, and conclude with a summary of the main contributions of this thesis

1.1 Thesis Contributions

The present thesis contributes to motion sensing research by presenting novel techniques for processing and refining sensor measurements collected from mobile devices, and by demonstrating the value of the proposed techniques with relevant use-cases. In the first part of this thesis we propose a collection of algorithms for estimating the attitude of the device by isolating

the gravity component within the three-dimensional accelerometers measurements. We separately consider gravity component estimation while relying solely on accelerometers, and when gyroscope measurements are available. Our approaches are specifically aimed to provide more robust gravity component estimation, and thus capture of linear acceleration, during situations where the device is influenced by periods of sustained linear acceleration. To quantify the captured linear acceleration in a way that can be efficiently used for machine learning-based motion inference systems, we then propose a set of peak and segment features. These features are designed for quantifying key aspects of sustained linear acceleration events, while also maintaining intuitively clear correspondence to the underlying physical phenomena. To demonstrate the effectiveness of the techniques presented in the first part of this thesis, we detail an accelerometer-based transportation mode detection system capable of discerning between fine-grained automotive modalities, while only relying on measurements about a device-embedded accelerometer. The contributions presented in the first part of this thesis are based on Article I, and Article II.

The second part of this thesis investigates collecting detailed information about users' movement and activities within a target space. As a key conceptual contribution, we study inverting the problem of generalization within motion sensing systems. Specifically, instead of aiming to design a system able to generalize over the myriad of variables related to motion sensing on mobile devices, we investigate instrumenting researchers with a prefixed configuration of sensing devices, and explore how to optimize performance when such configuration can be ensured. We then present our approach for generating enriched trajectories, which combine spatiotemporal movement trajectory, details on physical motion, and manual annotations. To demonstrate the effectiveness of these approaches, we present a method, *crowd replication*, for quantitative evaluation of public spaces. As the key technical contribution within crowd replication we detail construction of pedestrian models, which learn a personalized model for step and pace detection for a given user-placement combination. The contributions presented in the second part of this thesis are based on Article IV and Article V.

Finally, in the third part of this thesis we present two example applications for motion sensing. The first application, *Matkahupi*, is a persuasive mobile application promoting eco-friendly transportation based on automatically detected transportation behavior. Studies related to Matkahupi contribute by identifying key design challenges related to design, implementation and usage of automated transportation mode detection for promot-

ing behavior change towards more eco-friendly transportation. The second example application, *CeeSR*, is a collaborative speech monitoring system on smart devices, which opportunistically leverages co-located devices for energy-efficient speech monitoring. Within CeeSR, we contribute to motion sensing research through design and implementation of a motion sensing submodule that detects changes in physical device configuration, which can alter the audio information captured by the participating devices. The contributions presented in the third part of this thesis are based on Article III and Article VI.

1.2 Contributions of the Author

Article I. The concept of leveraging horizontal linear acceleration for transportation mode detection, and the resulting peak and segment features are contributed by the present author. The concept and algorithm development of the gravity estimation algorithm is joint work with Petteri Nurmi. The hierarchical machine learning design and the overall implementation of the system are due to the present author. The author was responsible for organizing and conducting the data collection scenarios, and implemented the related Android applications. The evaluation setup and implementation, as well as writing of the article are joint work with the co-authors.

Article II. The original concept, algorithm development and implementation of the overall system are contributed by the present author, with valuable feedback from Petteri Nurmi in particular for the orientation change detection submodule. Evaluation of the system and writing of the article are joint work with Petteri Nurmi and Sasu Tarkoma.

Article III. The design of the transportation mode detection module used in the persuasive mobility application is contributed by the present author. The implementation of the module within the Matkahupi-application is joint work with Miika Sirén. Writing of the article is joint work with the co-authors.

Article IV. The concept and the original idea of crowd replication are due to the present author. The technical development and implementation of the related software and algorithms are conducted by the present author. Evaluation of the overall system, conducting the field study, interpretation of the results and writing of the article are joint work with the co-authors.

Article V. The concept and technical development of pedestrian models are contributed by the present author. The evaluation of the overall system, field study, interpretation of the results and writing of the article are joint work with the co-authors.

Article VI. The original concept of collaborative sensing approach is joint work with Petteri Nurmi. The design and algorithm development of the motion sensing component within the work is contributed by the present author. Evaluation of the system, interpretation of the results and writing of the article are joint work with the co-authors.

Chapter 2

Motion Sensing

In this chapter we provide an overview of motion sensing on mobile devices and its main challenges. In Section 2.2 we provide a concise summary of the operating principles of accelerometer and gyroscope sensors, and define notation used throughout this thesis. In Sections 2.4 and 2.5 we make a distinction between *physical motion sensing* used to track physical properties, such as velocity and heading, of the device, and machine learning-based *motion inference systems* used to map motion information with activities and physical phenomena. Finally, in Section 2.6 we present transportation sensing as a representative application field for motion sensing techniques, and provide a summary of work related to transportation mode detection.

2.1 Motion Sensing on Mobile Devices

The ability to passively capture motion information about the movement of people and objects provides a key building block for various context sensing applications on mobile devices. The underlying physical attributes of motion information, i.e., acceleration and angular velocity, can be used directly to track the velocity, orientation and heading of the device. When integrated over time, such information can be used to track the relative location of the device through techniques such as inertial navigation (INS) [49] and pedestrian dead-reckoning (PDR) [78]. However, within mobile devices the motion sensors are typically unaligned with Earth-centered coordinates, thus requiring first transforming sensor measurements to a global coordinate system [150]. During the transformation process, the three-dimensional accelerometer measurements, commonly augmented with gyroscope measurements about angular velocity, are used to estimate device inclination respective to the gravity vector [95]. Besides acting as an in-

termediate step for reference frame transformation, the determined device inclination can be used to, e.g., adjust user-interface within mobile devices [54], as an interaction mechanism within mobile games and virtual reality systems [64], or to detect specific activities [7]. Physical motion sensing can additionally enable quantifying the magnitude of linear acceleration, usable for, e.g., monitoring injury risk [87] and to detect dangerous vehicle maneuvers [58].

Another, widely-used approach to leverage motion information is to employ machine learning-based motion inference methods to map motion information with physical activities and phenomena. Instead of using raw sensor measurements, these systems typically employ a set of processing and refining steps, referred to as *sensing pipeline*, to abstract motion information into a format suitable for machine learning algorithms. The sensing pipeline typically consists of a preprocessing step, transformation into suitable motion representation, and feature extraction stage [16]. Machine learning-based motion inference has been widely applied across several research domains. For instance, the activity recognition domain includes a wealth of studies investigating different combinations of sensors and activities [7, 72]. Typical activities of interest include movement-related activities, such as walking, running, biking, sitting or standing, and everyday activities, such as eating, drinking, reading or exercising [7]. In the closely related transportation domain, motion inference can be leveraged for transportation mode detection [116], tracking users' transportation routines [98], to measure users' transportation-related CO_2-footprint [38], and amount of physical exercise during transportation [27]. More elaborate motion inference techniques combining physical motion sensing and machine learning-based motion inference can reveal details about driving safety [58], road conditions [106] and fuel-efficiency of transportation [42].

Health related applications can utilize physical motion sensing for, e.g., fall detection [14] to monitor persons with a heightened risk and consequences for falling. Activity recognition can help clinicians and practitioners to assess physical activity of patients and prescribe timely interventions to support physical well-being and recovery [74]. A more detailed analysis of motion information about users' gait and walking can reveal early signs of neurodegenerative movement disorders, such as Parkinson's disease [6]. Closely related to the health domain, the wellbeing and sports domain can leverage motion information for automated exercise tracking, and performance assessment [50, 70, 117]. Within the security domain, motion sensing has been shown to be able to passively sense and reconstruct user's transportation routes, revealing privacy-sensitive transportation rou-

tines [98]. In more benevolent security-related applications, motion information can be leveraged to allow gesture-based user authentication [154], and for continuous authentication systems based on biometric details about users' gait [39]. In addition to the specific domains, motion information is frequently used to support other functionalities. For instance, motion information can be utilized to enhance location tracking accuracy [131], implement energy-efficiency mechanisms [9], monitor physical configuration of devices in collaborative sensing systems [76], and to implement motion compensation systems for sensors that can be ill affected by motion [48].

2.1.1 Benefits and Challenges

As discussed above, the benefits from developing a sensing system on existing commercial mobile devices are manifold. Mobile devices are naturally carried by users throughout their daily lives, allowing continuous inference of context information when it is most relevant for the user. Other important benefits include the available computing, storage and networking capabilities present on the device, allowing efficient processing of the collected measurements, and communicating the information to other services. The potential user base for new mobile applications running on any of the major operating systems can scale from millions up to even billions of users, with efficient and trusted distribution channels and commercialization tools readily available on the device [3, 45].

The main challenges for effective motion sensing in general relate to motion sensors measuring the combined inertial forces affecting the device. Humans often perform multiple activities at the same time, while the underlying environment (e.g., transportation vehicle) can also influence the overall motion information. Consequently, distinguishing relevant motion information pertaining to activities of interest effectively generates a source separation problem. Additionally, motion sensors are not perfect at measuring actual physical forces affecting the sensors, requiring that noise and drift within sensor measurements are addressed [64].

Motion sensing systems operating on mobile devices are affected by unconstrained placement of the device, unknown and dynamic device orientation, and differences in quality and availability of sensors on target platform(s) [128]. Placement of the device i.e., where the user is holding the phone, has a direct effect on the measured motion information. The three main sources for placement-dependent variations for motion sensing are due to:

1. Human biomechanics during physical activities, which result in dif-

ferent parts of the human body undergoing different motion patterns during physical activity.

2. Firmness of the placement, i.e., how much space the device has for moving separately from the user's movements and thus producing movement information unrelated to the physical activity.

3. Signal attenuation from padding within a placement which can significantly dampen high-frequency components of motion information.

Orientation of the device is similarly unknown and dynamic, requiring transforming motion information into a representation that is robust to changes in sensor orientation. Finally, there are significant hardware and software variations within mobile phones, in particular within smartphones [128]. The hardware variations include differences in available sensor modalities, differences in sensor sampling frequency, different unit systems used to convey motion sensor measurements, and differences in quality and calibration of motion sensors. Software variations include different versions of operating systems, which can limit or prohibit different functionalities of an application. Additionally, other applications operating on the device can compete for the limited resources available on the device, or interact in unforeseen ways with the developed sensing application, for example through use of common sensors.

To address these challenges, researchers have investigated motion inference algorithms which could perform robustly regardless of sensor orientation or placement [83, 116], estimate body position placement of the device [66], and track orientation of the device [64] within global coordinate frame. Errors due to varying sensor quality can be mitigated through virtual recalibration, and different unit systems can be aligned by converting the measurements into a common unit system [82, 128]. Within this thesis, we focus on measuring linear acceleration within the presence of sustained linear acceleration, which is a key enabler for transportation-related applications. We also investigate enhancing the accuracy of motion inference methods during situations when a specific configuration of sensors and placements can be ensured, e.g., when researchers are instrumented with sensing devices.

2.2 Motion Sensors

Motion sensors refer to a family of sensors capable of measuring different aspects of motion. Motion sensor measurements provide low-level information about the combined forces affecting the device, discretized to intervals

determined by the sensor's sampling frequency. The employed sensors determine the number of degrees of freedom (DoF) of the available motion information. Commonly, motion information can be sensed with either 3, 6 or 9 degrees of freedom with accumulative availability of three-dimensional accelerometer, gyroscope and magnetometer.

Practically all modern off-the-shelf smartphones feature one or more accelerometers, supporting 3-DoF accelerometer-based solutions able to monitor the effect of combined forces on the device's acceleration within three-dimensional local coordinates. A majority of modern smartphones additionally include a gyroscope, enabling 6-DoF systems able to monitor rotation and acceleration of the device separately. Magnetometers, while strictly not measuring motion, have a close relationship with motion sensors as they are able to resolve direction of acceleration and rotation respective to the Earth's magnetic field. Within the scope of this thesis, however, we reserve notation of motion sensors to such sensors that can directly capture physical movement of the device, i.e., accelerometer and gyroscope. Below we provide a short introduction to the operating principle and key performance specification for accelerometer and gyroscope sensors. For an in-depth view of the history, hardware and technical details of these sensors, we refer the interested reader to [69, 103].

2.2.1 Accelerometer

Accelerometer is one of the most common types of motion sensors, measuring the acceleration produced by the combination of physical forces influencing the sensor. The underlying operating principle of a typical accelerometer is to monitor small seismic-mass, also referred to as proof-mass, suspended by a mechanism such as a spring or beam; see Figure 2.1. When acceleration is applied on the seismic-mass, the mass is displaced by an amount relative to the influencing acceleration, which can then be measured to yield the force of the exerting acceleration. Modern accelerometers generally found within mobile devices are Micro Electromechanical System (MEMS) devices. These devices are small enough to be realized in a monocrystalline silicon using surface micro-machining, and typically consist of a cantilever beam attached to a seismic-mass. As a relatively simple sensor, accelerometers are both affordable to fabricate and low power to operate with a standard accelerometer drawing approximately 15 – 20mW in active state [15]. The first MEMS accelerometers were designed to trigger crash systems controlling airbags [69], but have since expanded to a plethora of other domains such as stabilization systems, segways, pedometers, robotics and even household items such as washing machines [22].

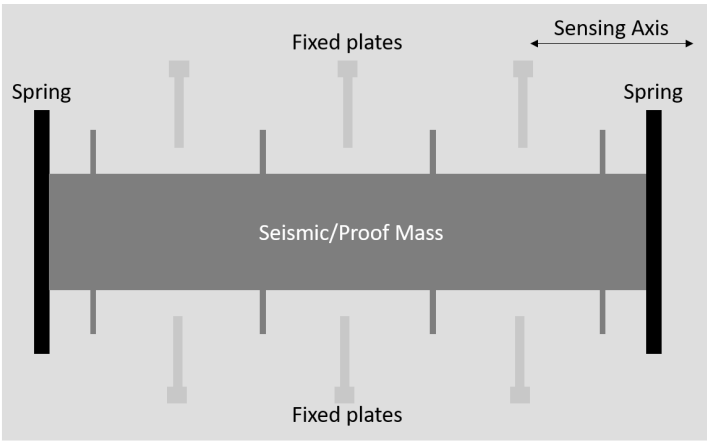


Figure 2.1: Illustration of the schematic design of a MEMS Accelerometer structure.

Within smartphones, accelerometers have evolved from simple tilt-sensors enabling adjusting screen orientation to instruments of automatic activity inference systems [71].

Accelerometers are not perfect at measuring the true acceleration forces affecting the sensor. In particular, low-cost accelerometers embedded within mobile devices can contain significant measurement inaccuracies [12]. Below we provide a (non-exhaustive) list of the errors and hardware specifications commonly reported for accelerometers:

- *Measurement range and sensitivity*, which specifies the maximum acceleration force that the sensor can measure (e.g., $\pm$ 2g), and the smallest acceleration that the sensor is able to distinguish.
- *Zero-G Bias*, which specifies the measured acceleration when there is no true acceleration present, e.g., during free fall.
- *Nonlinearity*, which specifies the scalar error applied on any measured true acceleration.
- *Package Alignment Error*, which specifies the misalignment between the sensor's sensing axes and the local device axes of the containing device.
- *Inter-Axis Alignment Error*, which refers to the extent of non-orthogonality between sensor axes on the sensing chip.

- *Cross Axis Sensitivity*, which specifies the amount of leakage between sensor axes, i.e., the amount of acceleration produced on the other axes when acceleration is exerted along the direction of one axis.
- *Accelerometer Noise*, which refers to the random deviation from the true acceleration, caused by the active electronic elements of the sensor and the containing device.

Acceleration and Notation

Within this thesis, we assume acceleration is expressed in m/s^2. An alternative popular unit to measure acceleration is to use a unit proportional to gravity g. We make a distinction between *total acceleration* and *linear acceleration*, where total acceleration is the sum acceleration from all the different sources affecting the sensor, and linear acceleration is a gravity-eliminated version of the total acceleration. Linear acceleration is further subdivided into *sustained*, *transient*, and *high-frequency* linear acceleration. Sustained linear acceleration refers to linear acceleration that persists over multiple seconds and is most commonly caused by mechanized movement. Transient linear acceleration refers to linear acceleration that has a brief, typically sub-second, duration, and is usually caused by human motion. Finally high-frequency linear acceleration refers to acceleration which occurs at low amplitude and frequencies exceeding 10Hz. Sources of high-frequency linear acceleration include, e.g., vibrations and jitters caused by engine resonance, or white noise in sensor measurements.

We denote three-dimensional accelerometer measurements with $A_F^{(x,y,z)}$, where superscript denotes the sensor axis, and the subscript denotes reference frame F within which the measurements are expressed. Gravity-eliminated versions of accelerometer measurements are notated with an accent $\acute{A}_F^{(x,y,z)}$. Similarly, true acceleration is denoted with a_F and true linear acceleration is denoted with $\acute{a}_F$. Horizontal linear acceleration on a plane perpendicular to the gravity component is notated as $\acute{a}_G^h$, where G denotes global reference frame, and corresponds to (x,y)-axes of gravity-eliminated accelerometer measurements rotated to global reference frame $\acute{A}_G^{(x,y)}$. Vertical linear acceleration on a vector aligned with the gravity component is denoted as $\acute{a}_G^v$ and corresponds to the z-axis of gravity-eliminated accelerometer measurements rotated to global reference frame $\acute{A}_G^z$.

A common procedure in applications that require precise acceleration measurements is to perform virtual recalibration of the accelerometer [83]. At its simplest form, virtual recalibration estimates the zero-g component and nonlinearity, often referred to as the offset and the scaling factor re-

spectively. The calibration is performed separately for each of the sensor axes. Formally, this corresponds to solving equation

$$a^{(x,y,z)} = K^{(x,y,z)} * A^{(x,y,z)} + b^{(x,y,z)} \tag{2.1}$$

for each of the sensor axes (x,y,z), where a is true acceleration, A is the measured acceleration, K is the scaling factor corresponding to nonlinearity, and b is the constant offset corresponding to the zero-g bias.

2.2.2 Gyroscope

Compared to accelerometers, gyroscopes are slightly more complex systems that measure the angular rate experienced by the sensor. Since the invention of the first gyroscope in 1852 by Leon Foucault, several types of gyroscopes with different operating principles have been investigated. Roughly categorized, gyroscopes can be divided into mechanical, optical, and MEMS gyroscopes [103]. Within the scope of this thesis, we focus on the MEMS gyroscopes as they are predominately used within mobile devices. Similar to accelerometers, MEMS gyroscopes are both cost-efficient to fabricate and low power to operate, consuming approximately 20mW when active [15].

The common operating principle of a mobile device-embedded MEMS gyroscopes is to leverage Coriolis acceleration to measure the angular rate experienced by the sensor. Formally, Coriolis force F_C experience by an object with mass m, velocity v, and angular rate Ω is

$$F_C = ma_c, \tag{2.2}$$

where a_c is Coriolis acceleration

$$a_c = 2 * (v \times \Omega). \tag{2.3}$$

Reflecting their operating principle, such gyroscopes have been coined as the Coriolis Vibratory Gyro (CVG) [29]. To demonstrate the operating principle of a CVG Gyroscope, consider the system illustrated in Figure 2.2. The system contains a proof-mass resonating along local the Y-axis, while being suspended by mechanical structures such as springs or beams. As the proof-mass resonates along the Y-axis, any rotations of the underlying surface that the gyroscope is mounted on will produce Coriolis acceleration along the system's local X-axis, measurable by the Coriolis Sense Fingers.

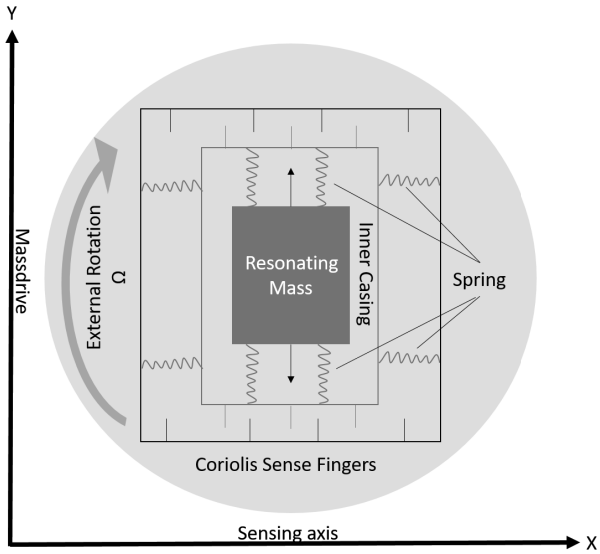


Figure 2.2: Illustration of the schematic design of a MEMS CVG Gyroscope structure.

s

The displacement x of the proof mass as a function of time t along the Y-axis is given by

$$y(t) = A_y \cos(\omega_y t), \tag{2.4}$$

where A_y is the amplitude and ω_y is the frequency at which the proof mass is vibrating along the Y-axis. When the system is rotated with angular rate Ω parallel to the (X,Y)-plane, the Coriolis acceleration caused along the X-axis equals

$$a_c = -2\Omega A_y \omega_y \sin(\omega_y t), \tag{2.5}$$

which then causes the proof mass to resonate along the X-axis proportionally to the applied angular rate Ω, which can then be used to measure the underlying angular velocity.

Quality and calibration of gyroscopes are commonly quantified using five error metrics [103]:

- *Angle Random Walk* (ARW), which refers to the random white noise in the measurements caused by the active electronic elements in MEMS gyroscopes.

- *Offset error*, also called the zero rate level, which refers to the near-constant error (i.e., offset from zero) along each of the sensor axes when the device is in rest. Offset error is typically the most significant error source in gyroscopes, which can result in rapid accumulation of error if not addressed correctly.

- *Offset instability*, which refers to the random drift of the offset error, i.e., deviation from a constant value in stable environment. Offset instability is a critical specification in gyroscopes, since constant offsets are straight-forward to identify and compensate through virtual recalibration, but any significant instabilities in the offset value render the calibration process more complex and error-prone.

- *Temperature sensitivity*, which refers to the sensitivity of the sensor's noise, offset and scalar factor to changes in temperature. All low to moderate quality MEMS gyroscopes are naturally sensitive to temperature changes. However, most modern gyroscopes include an integrated temperature sensor on the chip, which can handle temperature compensation on the hardware level.

- *Shock and vibration sensitivity*, which refers to the sensor's ability to track true angular velocity in situations where the sensor is subject to vibration or shock input. Robustness against vibration is of particular importance when tracking vehicles, which typically contain high frequency motion resulting from a combination of engine noise and surface-to-vehicle contact.

Angular velocity and Notation

Within this thesis, we assume that angular velocity measured by gyroscopes is expressed as radians per second rad/s. Other popular units include degrees per second, and revolutions per second. Analogously to accelerometer notation, we note three-dimensional gyroscope measurements as $\Omega_F^{(x,y,z)}$ and true angular velocity as ω_F, where the superscript denotes the sensor axis, and the subscript denotes the reference frame of the measurements. Gyroscopes are not affected by the gravity pull, and thus similar separation of linear and true rotation is not required.

We note that, angular velocity measured by a gyroscope axis should be interpreted as rotation *around* the respective axis. For instance, turning along a horizontal plane perpendicular to the gravity component (e.g., a vehicle turning along a road) is captured by rotation around the z-axis of a gyroscope aligned with global reference frame Ω_G^z.

2.3 Context Inference using Motion Information

In order to recognize and quantify the physical phenomena producing the motion sensor data, the measurements need to be passed through a number of data processing and refinement steps. The required data processing steps depend on the main principle of how the motion data is leveraged. Within the scope of this thesis, we define two such categories: *Physical motion sensing* and *Motion inference systems*, defined as:

(i) Physical motion sensing refers to systems which leverage the physical basis of the motion information to track and measure movement and orientation of the device.

(ii) Motion inference systems refer to systems which interpret the motion information by leveraging machine learning methods to recognize patterns and qualities within the sensor measurements, typically expressed using *features*, and map these with ground truth labels associated with sensor measurements.

We note that in practice many systems exhibit qualities of both of these categories, e.g., by constructing features, such as velocity change or sharpness of turning maneuver, that have direct physical correspondence. In the next section, we focus on methods related to physical motion sensing and the steps required to refine and transform the raw sensor measurements in order to measure and quantify physical movement of the user. We return to methods related to motion inference systems using machine learning in Section 2.5.

2.4 Physical Motion Sensing

Physical motion sensing refers to the process of refining and transforming raw motion sensor measurements into motion information usable for tracking and quantifying the physical movement of the user. The first step for physical motion sensing systems is to *preprocess* the measurements in order to remove systematic errors and noise, and thus increasing the correspondence of the measurements to the physical phenomena generating the data. Next, the measurements are typically transformed from the device's local coordinate system into Earth's coordinate system using *reference frame transformation*. Finally, the motion information can be *decomposed* into components capturing different aspect of the movement. Below we provide a summary of each of these steps.

2.4.1 Preprocessing Motion Sensor Measurements

One of the most common preprocessing techniques is to employ a digital filter to reduce noise and jitter from the data. High-frequency noise can be cost-effectively filtered using a simple low-pass filter based on, e.g., averaging the sensor measurements, Fourier transformation, or discrete Wavelet decomposition [159]. While beneficial for improving the signal-to-noise ratio of the sensor data, the employed filter needs to be carefully designed to avoid losing relevant information for the target application, in particular when using the high-frequency motion spectrum [88]. In applications which require more accurate separation between noise and true motion signal, more elaborated filter designs based on, e.g., Singular Value Decomposition, Savitzky-Golay Filter, or adaptive Kalman Filters can be applied at the cost of extra computational complexity [79].

The preprocessing stage can also include virtual recalibration of the motion sensors to mitigate systematic errors in the sensor measurements due to calibration and hardware errors; see Sections 2.2.1 and 2.2.2 for details of the different calibration error types for accelerometer and gyroscope respectively. Virtual recalibration of an accelerometer can be performed with either active (user-driven) or passive (automatic) procedures [83]. The common parameters to estimate for accelerometers are the zero-g bias b, i.e., the constant error offset when there is no actual acceleration affecting the device, and sensor sensitivity estimated by the scaling factor K, i.e., the scalar offset between measured and true acceleration. These calibration parameters can be readily estimated using acceleration force from gravity as a reference, i.e., assuming that the squared norm of total acceleration equals 1g when the device is in rest. By collecting a set of measurements in sufficiently varied device orientations during a (near) static inertial state, an estimation procedure, e.g., linear minimum variance unbiased estimator [105], can be used to estimate parameters b and K. For further details and formulation of the calibration procedure, coined *quasi-static moments detector* approach, we refer the reader to Lötters et al. [82].

Virtual recalibration of gyroscopes within mobile devices typically involves estimating only the offset error b, i.e., the near-constant angular velocity measured when the device is in a static inertial state. Accurately estimating the scalar offset K for gyroscopes is generally difficult without specialized equipment as there is no reliable static reference, analogous to gravity for acceleration, for angular velocity. Since the scalar offset is additionally generally small for gyroscopes, the offset is typically not considered for virtual recalibration of gyroscopes [44].

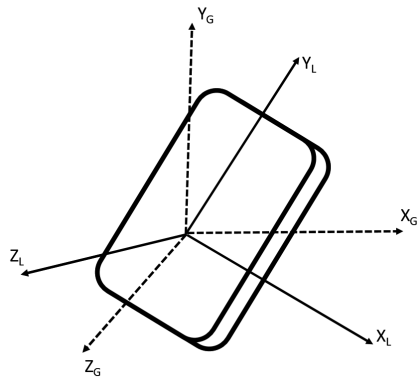


Figure 2.3: A device and its local body frame (X_L, Y_L, Z_L) in comparison to the global inertial reference frame (X_G, Y_G, Z_G).

2.4.2 Reference Frame Transformations

An important distinction, relevant in particular for mobile devices, is that the motion sensors provide measurements respective to their local body frame. Consequently, the sensor measurements do not directly relay movement information respective to the global inertial reference frame; see Figure 2.3 for an illustration of the two different reference frames. Within the scope of this thesis, we refer to the device reference frame as the *Local frame* L, and the Earth Centered Inertial frame (ECI) [29] as the *Global frame* G. The reference frame is notated with a subscript and transformation between coordinate systems with an arrow between the corresponding frames. Thus, transforming motion sensor measurements m from local body frame to ECI is noted as $m_{L \to G}$.

Orientation Representation

In general, there are three commonly used mathematical constructs for representing orientation of an object within three-dimensional space: Euler angles, quaternions, and direction cosine matrix (DCM) [31]. Each of these approaches have their unique advantages and limitations and are transformable between each-other. Within the scope of this thesis, we use the Euler representation of orientation as it provides intuitively easy tools to describe orientation and coordination transformations. We note that actual implementations of any of the presented orientation tracking or transformation techniques within this thesis can use any available orientation representation systems.

Euler angles define orientation of an object using three consecutive rotations around the coordinate axes of a reference frame: $\boldsymbol{R_x}$ called the *roll* (ϕ) around the x-axis, $\boldsymbol{R_y}$ called the *pitch* (θ) around the y-axis, and $\boldsymbol{R_z}$ called the *yaw* (ψ) around the z-axis, where $\boldsymbol{R}$ denotes a rotation matrix. A rotation matrix of dimension [n × n] can be used to transform a [n × m] matrix from one coordinate system to another while maintaining its internal geometry unchanged. There are three main conventions used for the sequence of rotations, i.e., xyz, zxz and zyz, of which we here use the first one xyz: $\boldsymbol{R} = \boldsymbol{R_x}\boldsymbol{R_y}\boldsymbol{R_z}$, where

$$\boldsymbol{R_x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{2.6}$$

$$\boldsymbol{R_y}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2.7}$$

$$\boldsymbol{R_z}(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.8}$$

The main limitation of Euler angles is the presence of singularity points, called *gimbal locks*, which can cause loss of a degree of freedom. Gimbal locks occur when two or more gimbals representing the Euler angles are set in parallel configuration. As an example, gimbal lock occurs when the device is in upright position, i.e., pitch $\theta = 90°$ and thus $\cos(\theta) = 0$, rendering roll and yaw indistinguishable. However, within this thesis we consider motion sensing using accelerometer and gyroscope, making the rotation $\boldsymbol{R}_z(\psi)$ linking the z-axis of local and global reference frames inherently unsolvable as there is no static reference for solving ψ. In practice we are thus limited to transforming measurements only regarding the device *inclination* Θ

$$\Theta = \begin{bmatrix} \phi \\ \theta \end{bmatrix}, \tag{2.9}$$

alleviating the problems of gimbal locks. Using the defined notation, rotating motion sensor measurements m from local to global frame inclination can be expressed as:

$$m_{G(\Theta)} = m_{L(\Theta)\rightarrow G(\Theta)} = \boldsymbol{R(\Theta)}m_L \; = \boldsymbol{R_x}\boldsymbol{R_y}m_L \; . \tag{2.10}$$

2.4.3 Gravity Component Estimation

Within mobile devices with unconstrained device orientation, automatic reference frame adjustment commonly relies on *Gravity component estimation*, i.e., estimating the distribution of gravity component $G = (g_x, g_y, g_z)$ within the three-dimensional accelerometer measurements [95]. Estimation of the gravity component allows resolving roll ϕ and pitch θ angles between the local and global reference frames:

$$\tan(\phi) = \frac{-g_x}{g_z} \tag{2.11}$$

$$\tan(\theta) = \frac{g_y}{\sqrt{g_x^2 + g_z^2}}. \tag{2.12}$$

The gravity component can be estimated directly from the accelerometer measurements, or through sensor fusion of accelerometer and gyroscope. Below we detail existing methods for both groups.

Accelerometer-based Gravity Component Estimation

Gravity component estimation solely from accelerometer measurements is inherently a heuristic process, as the linear acceleration and changes in distribution of the gravity component within the sensor axes cannot be reliably decoupled. The estimation is generally accurate when the device is at rest, and decreases in accuracy as variation of measurements increases. Quality and calibration of the sensor additionally have an impact on the accuracy of the gravity component estimation, in particular in the case of low-grade sensors frequently embedded within mobile devices [12]. A widely adopted approach for estimating the gravity component within accelerometer measurements is to filter measurements over a data frame of fixed duration. Commonly used filter designs include averaging of the measurements using mean [95] or median [96], which can yield reasonable estimation of the gravity component given a sufficiently long data frame. Various durations for the data frame have been employed in the mobile sensing domain, e.g., the originating work from Mizell [95] estimates the gravity component by averaging measurements over 30 seconds, Mohan et al. [96] use a median over 10 seconds, Lu et al. [83] use a mean over 4 seconds, and Wang et al. [143] use a mean over 8 seconds. A more dynamic gravity estimation can be achieved by employing a smoothed mean calculated for each sample individually [61]. An alternative filter design is to employ a low-pass filter by converting the signal into time-frequency domain through, e.g., Fourier Transformation, and retaining only the low frequency components [98].

However, in addition to being sensitive to the used data frame length, frequency filtering requires careful selection of the frequency cut-off in order to correctly separate linear acceleration and the gravity component.

Filtering accelerometer measurements can generally yield a good estimation of the gravity component within situations where gravity is the only sustained force exerting the accelerometer, and variation of acceleration is evenly distributed around the gravity component. This assumption, however, is seldom valid within the mobile domain, and especially during sustained linear accelerations caused by mechanized movement. As an example, consider Figure 2.4 illustrating acceleration along one of the three sensor axes during a bus transit, and estimation of the gravity component with different filtering techniques and parameters. Periods of sustained linear acceleration are evident in the underlying acceleration (black line in the figure) as peaks in the acceleration. As evident from the figure, the longer the data frame, the better the gravity estimation can avoid tracking the linear acceleration peaks. However, longer data frames suffer from latency in tracking the actual gravity component, in particular during a shift in the device inclination, as evident near the 1000-second mark.

An alternative approach to gravity component estimation is to monitor accelerometer variance and magnitude, and estimate gravity opportunistically during near-static periods when the main force affecting the accelerometer is gravity. For instance, Kunze et al. [67] monitor variance of acceleration and the magnitude of total acceleration, and estimate gravity over a short data frame whenever variance is small and magnitude of total acceleration is near 1g. Figure 2.5 illustrates the gravity estimation using the same example data from a bus transit. The opportunistic gravity estimation approach can capture periods of sustained linear acceleration almost perfectly when there is a recent gravity estimate, and device inclination remains relatively unchanged. However, the approach has three key limitations. First, selecting a suitable variance threshold value to trigger updating gravity estimation is highly susceptible to hardware variation, the underlying motion environment, and noise characteristics due to, e.g., engine vibration and vehicle-to-surface contact. Consequently, any constant threshold value struggles to generalize over a larger deployment, heterogeneous devices, or use-cases. Second, using an additional filter requiring that the norm of the accelerometer measurements is near 1g requires an accurately calibrated accelerometer, which contrasts with the frequently used low-quality and poorly calibrated sensors available on mobile devices. As an example, consider an acceleration period of mechanized movement with a typical linear horizontal acceleration of $1.5m/s^2$. While notable

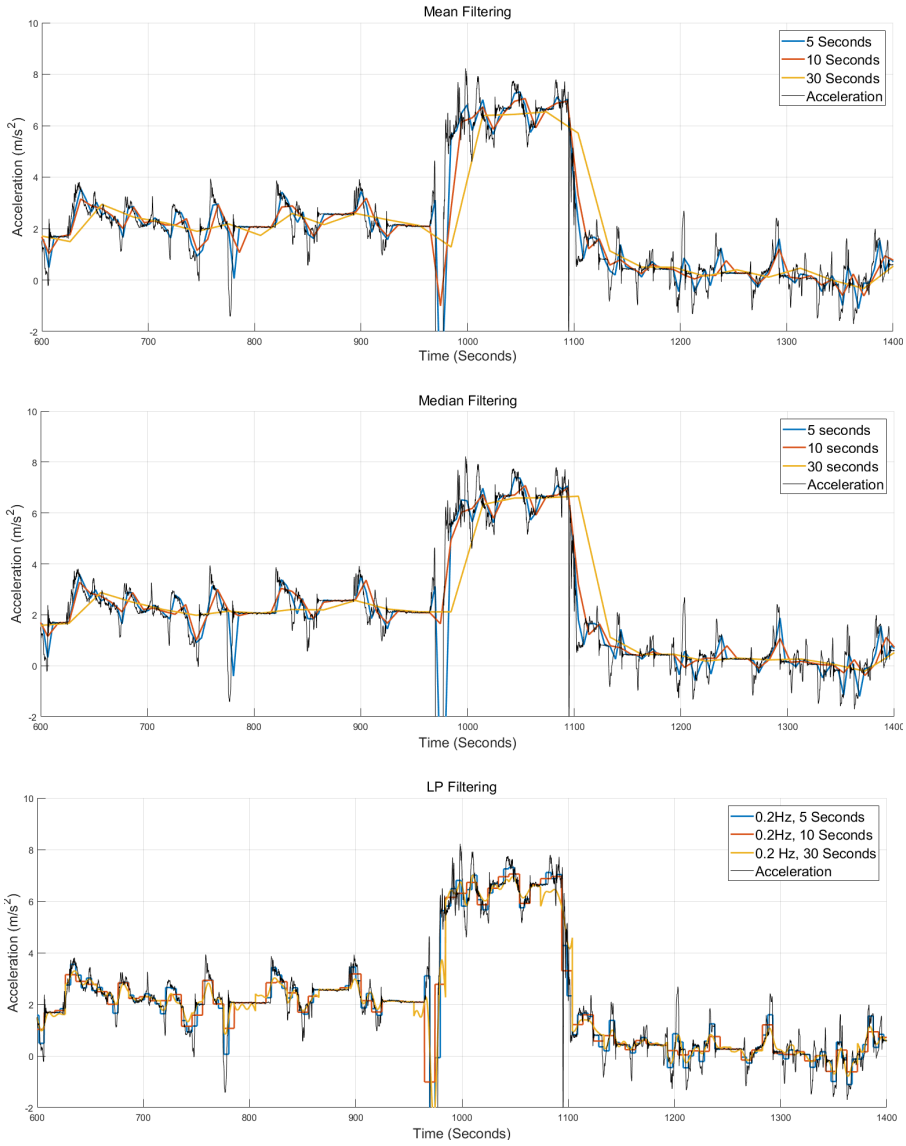


Figure 2.4: Overview of gravity estimation using filtering with different techniques and parameters during a bus transit. Acceleration (black) is measured by one of the three accelerometer axes, most aligned with the direction of movement, and filtered using a 10ms centered median filter for better readability.

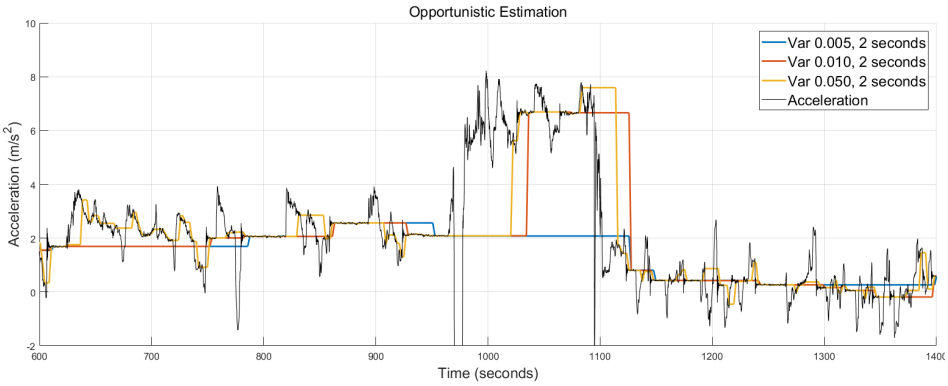


Figure 2.5: Opportunistic gravity estimation during a bus transit.

acceleration with respect to the vehicle's forward velocity, such acceleration only increases the norm of overall acceleration from $9.81 m/s^2$ to $\sqrt{9.81^2 + 1.50^2} = 9.92 m/s^2$, i.e., by a mere 1.1%. Thirdly, the shifts in device inclination render current gravity estimate obsolete and thus risk producing large errors until a new gravity estimation is triggered. These inclination shifts are typically produced by user interacting with the device, standing up or sitting down, or shifting their posture, making them frequent within mobile devices [101].

IMU-based Gravity Component Estimation

Using the mobile device's embedded accelerometer and gyroscope to track three-dimensional acceleration and angular velocity corresponds to considering the device as an Inertial Measurement Unit (IMU). IMU is defined as a self-contained device which measures three-dimensional linear acceleration and three-dimensional rate of rotation affecting the unit respective of an inertial reference frame, commonly the Earth-Centered Inertial frame (ECI) [29]. A broader definition of IMU additionally includes measuring the magnetic field surrounding the device, which allows fully determining device orientation with respect to Earth's coordinate system. Within the scope of this thesis, such systems are distinct from the classical definition of IMU and coined as the *Magnetic IMU* or MIMU.

Gravity component estimation within IMUs can be considered a subproblem of the more general *attitude estimation*, i.e., determining orientation of the sensing unit with respect to the global reference frame. The problem has been widely studied within, e.g., robotics, sports, navigation

systems, medical rehabilitation, and augmented reality [64, 84]. Compared to pure-accelerometer-based approaches, the main benefit of IMU-based approach is the ability to decouple device rotations, measurable by gyroscope, from linear acceleration.

Combining accelerometers and gyroscopes for orientation estimation provides complementary measurements about tracking device's gravity component, i.e., accelerometers can provide a good estimate of the gravity component over a sufficiently long data frame, while gyroscopes can accurately track changes in device orientation over short data frames. The principal design concept used within IMU orientation estimation is based on taking advantage of this relation. Accordingly, accelerometer is used to offset gyroscope drift over time, while gyroscope is used to track orientation changes over a short time frame. The simplest implementation of this technique is called the *Complementary Filter* [25], formally defined as:

$$\Theta_t = \alpha * (\Theta_{t-1} + \omega * dt) + (1 - \alpha) * \Theta_t^{acc}, \tag{2.13}$$

where Θ_t is the inclination of the device at time t, ω is the rotation measured by gyroscope, α is a weight term balancing between gyroscope and accelerometer sensors, and Θ_t^{acc} is the inclination derived from accelerometer measurements at time t. A common range for α values between $0.97 - 0.99$, allocating most (i.e., $97 - 99$ %) of the weight to gyroscope, while accelerometer is mainly used to offset the gyroscope drift.

More elaborate formulations using a probabilistic framework are commonly achieved using (Extended) Kalman filter as a basis for the orientation estimation techniques [28]. However, these techniques typically require high-precision (M)IMUs, which contrasts with the lower quality sensors embedded within mobile devices. Additionally, the large state vectors typically required to estimate the rotational kinematics present a large computation overhead for the attitude tracking [84]. To address the computational overhead, Mahony et al. [85] and Madgwick et al. [84] have proposed computationally more light-weight approaches leveraging gradient descent algorithm and quaternions and as the basis for orientation representation. However, both of these techniques also break during sustained linear accelerations and thus are not able to provide accurate gravity estimation during mechanized transportation.

2.4.4 Decomposition of Motion Information

Estimation of gravity component $G = (g_x, g_y, g_z)$ serves a dual-purpose for interpreting motion information. First, estimation of the gravity compo-

nent establishes a reference point between local and global reference frames, enabling transforming sensor measurements m between the reference frames $m_{L(\Theta)\rightarrow G(\Theta)}$ respective to the inclination Θ, as described in Section 2.4.2.

The second benefit is that effects of gravity can be subtracted from the accelerometer measurements A_L to yield linear acceleration $\acute{A}_L$

$$\acute{A}_L = A_L - G = (A_L^x - g_x, A_L^y - g_y, A_L^z - g_z). \tag{2.14}$$

Linear acceleration measurements can then be transformed into global frame respective to inclination, yielding $\acute{A}_{G(\Theta)} = \acute{A}_{L(\Theta)\rightarrow G(\Theta)}$. Within $\acute{A}_{G(\Theta)}$ vertical linear acceleration can be directly read from the z-axis of the transformed accelerometer measurements, while acceleration on the horizontal plane is captured by the (x,y)-axes of the transformed accelerometer sensor:

$$\acute{a}_G^v \simeq \acute{A}_{G(\Theta)}^z \tag{2.15}$$

$$\acute{a}_G^h \simeq \acute{A}_{G(\Theta)}^{(x,y)}. \tag{2.16}$$

As detailed in Section 2.4.2, the sensor measurements are only transformed respective to inclination of the device $\boldsymbol{R(\Theta)}\acute{m}_L = \boldsymbol{R_x R_y}\acute{m}_L$. As a result, the horizontal motion information is unaligned with the yaw-rotation $\boldsymbol{R_z}(\psi)$ respective to the global reference frame, and thus $m_{G(\Theta)}^{(x,y)}$ captures horizontal movement in arbitrary (x,y) - coordinate alignment. Often, however, horizontal linear acceleration can be further decomposed into two orthogonal acceleration vectors, i.e., tangential and radial components:

$$\acute{a}_{G(\Theta)}^{h(t,r)} = \acute{a}_{G(\Theta)}^{h(t)} \cdot \acute{a}_{G(\Theta)}^{h(r)}. \tag{2.17}$$

Tangential linear acceleration $\acute{a}_{G(\Theta)}^{h(t)}$ is derivative of velocity v along direction of horizontal movement, i.e., the observed change in horizontal velocity of an object Δv. Radial linear acceleration $\acute{a}_{G(\Theta)}^{h(r)}$, also called centripetal or normal acceleration, is orthogonal to tangential acceleration and is caused by changes in the direction of velocity. Radial linear acceleration is associated with angular velocity (and thus gyroscope) with equation:

$$\acute{a}_{G(\Theta)}^{h(r)} = r\omega, \tag{2.18}$$

where r is the radius of curvature.

Most motion sensing applications running on mobile devices are primarily concerned with the tangential linear acceleration component as it relates to the device's velocity v at time τ through integration:

$$v(\tau) = v(\tau_0) + \int_{\tau_0}^{\tau} \acute{a}^{h(t)}_{G(\Theta)}(u)du \tag{2.19}$$

and to position s and distance d through integration of velocity:

$$s(\tau) = s(\tau_0) + \int_{\tau_0}^{\tau} v(u)du \tag{2.20}$$

$$d = s(\tau) - s(\tau_0) = \int_{\tau_0}^{\tau} v(u)du. \tag{2.21}$$

We note that accelerometer provides discretized measurements about acceleration at a frequency defined by the sensor sampling rate. Consequently, integration is typically performed using numerical integration, i.e., approximating the area under the tangential linear acceleration [36].

2.5 Motion inference using Machine Learning

Contrary to physical motion sensing, motion inference using machine learning does not directly rely on the physical properties of the sensor measurements. Instead, motion inference methods, on a general level, seek to approximate a function $f(x) = y$ between *input* x and *output* y, where x is the motion information sampled and refined from motion sensors, and y is the physical activity or phenomena the system is seeking to predict. On a general note, y can be of various forms, e.g., intensity of motion estimated by using regression, a specific activity inferred using classification, or a probability distribution over multiple considered classes.

The approximation, or learning, of the true function f is noted as $f^*(x;\theta) = \hat{y}$, where θ is a vector or parameters yielding best approximation of f, and $\hat{y}$ is the predicted output for input x given parameters θ. Within motion inference systems, the most common approach is to employ a supervised machine learning approach, depicted in Figure 2.6. In the figure, the key steps related to motion sensing techniques are the modules prior to model training, which refine the raw motion sensor data into usable input x for the machine learning algorithm. These modules, commonly referred to as the *sensing pipeline* consist of a sequence of *Preprocessing*, *Motion*

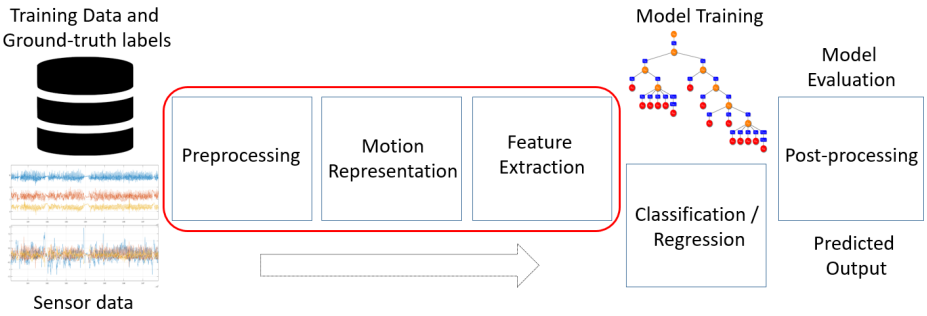


Figure 2.6: Illustration of the typical pipeline for supervised machine learning for motion information inference.

Representations and *Feature Extraction*. The Preprocessing module, aiming to correcting systematic errors within the raw sensor measurements and improve signal-to-noise ratio, is described in Section 2.4.1. The latter two stages are detailed in the following sections.

2.5.1 Motion Representations

A central technique for feature extraction is to first transform the motion sensor measurements into *motion representations* able to effectively and robustly convey motion information relevant for the target application. For instance, activity recognition methods for smartphones frequently utilize the L^2-norm, calculated as $\sqrt{x^2 + y^2 + z^2}$, over sensor axes (x,y,z) as the main representation [116, 123, 140]. Using the L^2-norm effectively corresponds to monitoring the overall magnitude of the motion instead of the three-dimensional distribution of the measurements, enabling orientation-agnostic feature extraction. While elegant and computationally light, the L^2-norm has an inherent weakness that it can be dominated by a strong source of acceleration. In particular, the L^2-norm of accelerometer is dominated by the gravity component which can easily mask more subtle accelerations parallel to the gravity component, making it unsuitable for applications requiring information about horizontal accelerations.

Motion representations can also be produced by finding basis vectors of the motion data using, e.g., principal component analysis (PCA), independent component analysis (ICA), factor analysis (FA), or non-negative matrix factorization (NMF), and projecting the measurements along the resolved basis vectors [126]. This procedure corresponds to transforming the data from the device's local reference frame to dynamically determined

coordinates minimizing co-variance of the sensor axes. However, since the procedure is typically repeated for every new data frame, there is a risk of substantial variation in the produced coordination, resulting in unstable motion representations as the distribution of motion fluctuates between the sensor axes.

An alternative method to characterize motion information is to employ a distribution-based motion representation. For instance, Plötz et al. [68, 110] propose using an empirical cumulative distribution function (ECDF) of whitened measurements about a data frame. The main limitation of distribution-based representations is that they ignore the temporal structure of the motion information. Kwon et al. [68] address this limitation by extending the ECDF-based approach with structural information using local region variability, scale variability, context-length variability, and wavelet variants. Closely related to distribution-based representations, motion sensor measurements can also be discretized into symbolic strings, allowing compressing the motion information, but potentially losing granularity and information [94]. Discretization is typically performed by defining a mapping between measurements ranges and corresponding symbols. More elaborated techniques, such as the symbolic aggregate approximation (SAX) proposed by Lin et al. [81], allows dynamically defining the measurement ranges so that they form roughly equiprobable bins of the underlying sensor data.

Effective motion representations can be produced by leveraging techniques from physical motion sensing. Such representations typically rely on estimation of the gravity component, as detailed in Section 2.4.3, to produce horizontal and vertical motion representation. Such motion representations can be particularly effective when most of the relevant motion information occurs on the horizontal plane, e.g., during transportation. In case information about direction of the movement is available, the horizontal motion representations can be further divided into tangential and radial acceleration respective to the movement direction of the user.

Finally, motion representations can be used in conjunction to produce representations increasingly specific to certain types of motion. For instance, other motion representation techniques can be applied on top of the gravity-eliminated horizontal or vertical acceleration. For example applying the L^2-norm on the horizontal acceleration can capture the combined acceleration forces affecting the velocity and direction of horizontal movement. PCA has been similarly applied on top of the ECDF representation of motion sensor data, producing motion representations suitable for activity recognition tasks, as demonstrated by Plötz et al. [110].

2.5.2 Feature Extraction

During feature extraction, the motion representations are transformed into descriptive *features* that capture key aspects of the motion information relevant for the target application. Note that some motion representations can be used directly as features, particularly those relying on alternative basis representations. For example the n-point ECDF distribution effectively conveys percentile information about the data frame distribution at n points. We note that some recent advances in deep learning domain [100, 137], are able to automatically learn features during the model learning process, and thus do not require a separate feature extraction stage.

In classical motion inference, motion representations are split into data frames of predefined duration and overlap, after which a vector of features is calculated from each data frame. The overall process of transforming motion information into features is called *feature extraction*, while the selection and design of the individual features is called *feature engineering*. The main advantages of using features include:

- Removing redundant information and compressing information by abstraction. Typically each feature corresponds to a certain aspect of the data frame, expressed as a single numerical value, e.g., mean or variation of the measurements. Using a vector of descriptive features instead of sensor measurements directly can significantly reduce the size of the input x, thus decreasing both the time required to learn f^* and the time required to perform predictions $f^*(x, \theta) = \hat{y}$.

- Improving generalizability of the learned model by designing features that are robust to noise and outliers in the motion sensor data, e.g., by using the interquartile range or median of the measurements.

- Fusing domain-specific expertise to the system by designing features which capture key motion information for the target phenomena or activities. For instance, the time between peaks caused by heelstrike can be used as a feature to estimate users' walking or running speed [21].

The most commonly used features for motion representations preserving the physical basis of the measurements can be categorized into *time-domain features*, and *frequency-domain features* based on the basic principle of the applied technique [36]. Features derived from symbolic string motion representations use their own specific *string-domain features*. Below we provide

a concise overview of typical features derived from each of these domains. For more detailed analysis of features, we refer to work by Figo et al. [36].

Time-domain features include basic statistical measures, such as mean and variance, of the respective data frames. Time-domain features can express envelope metrics of the frame through features such as minimum, maximum, (interquartile) range, mean and median of the data frame. The average of the accelerometer measurements can reveal orientation of the device, as detailed in Section 2.4.3. The distribution and stability of the measurements can be expressed, for example, through standard deviation, variance, zero (or mean) crossing rate, or pair-wise difference. Standard deviation and variance can additionally convey the overall intensity of physical motion. Using autocorrelation can reveal repetitive patterns within the signal, whereas cross-correlation can reveal linear relationships between different components of motion information.

Frequency-domain features are used to capture and quantify periodic patterns within the signal, which are typically generated by repetitive physical movement such as walking or cycling. In order to extract frequency-domain features, the signal first needs to be transformed into frequency spectrum using, e.g., Fast Fourier Transformation [145] or Wavelet decomposition [86]. Common frequency-domain features include key frequency bands of the considered activities, e.g., $0.5 - 3$Hz for pedestrian activities or over 10Hz for automotive modes [88]. Total energy of the signal can be used to estimate the energy expenditure of the movement. Spectral entropy captures the regularity of the signal, which can be used to help distinguishing between activities with similar energy level but different distribution of spectral coefficients.

String-domain typically relies on comparing the similarity of symbolic strings corresponding to data frames. The distance measure $D(A, B)$ comparing similarity between two equi-length symbolic strings A and B can be calculated using the Euclidean distance and a lookup table detailing distance between each pair of symbols. Closely related is the $MinDist$, which measures the lower bound of Euclidean distance between two symbolic strings. Other common distance measurements used to compare Strings include finding the longest common subsequence, different variations of the Edit Distance, and the Dynamic Time Warping (DTW) metric.

2.6 Transportation Sensing

In this section we focus on applications of motion sensing techniques within the transportation domain. Transportation has long been a popular domain for motion sensing applications as it simultaneously provides a challenging and versatile testing environment for motion sensing techniques, as well as enormous potential for commercialization and real-world application of successful techniques. The importance of advancing efficiency of transportation, and thus solutions supporting it, is emphasized by both the central importance of transportation networks within an urban environment, and the cost of transportation as functions of time, energy and CO_2-emissions [119, 146]. Research involving motion sensing within the transportation domain is here subdivided into three main branches: (1) Inertial navigation systems, (2) Monitoring transportation conditions, and (3) Transportation mode detection. Below we provide a concise summary of the first two research branches, after which we proceed with a more thorough review of the transportation mode detection field.

Within inertial navigation systems (INS), motion sensing is leveraged to estimate velocity and heading of the user. Using iterative integration of heading and velocity, the user's inertial state and relative location can be tracked while relying solely on the device-embedded motion sensors. The most common application of INS systems is to track users location when and where positioning based on external signal sources, e.g., GPS, WiFi, or GSM, is unavailable [131]. When applied for tracking pedestrian movement, these systems are referred to as *Pedestrian Dead Reckoning* (PDR) systems. A common approach is to leverage either INS or PDR systems in conjunction with other localization methods to increase location accuracy and/or decrease energy consumption of tracking users location [9].

The second branch, monitoring transportation conditions, focuses on detecting and quantifying conditions for vehicular transportation and assessing driving maneuvers during mechanized transportation. These systems typically leverage a mixture of physical motion sensing and machine learning techniques to measure both physical aspects of vehicular movement, and classification of the movement into predefined categories. Analysis of the driving maneuvers can be used to assess driver's skill as a function of driving behavior [58], and fuel-efficiency [42]. Automated detection of driving conditions using motion sensing include assessment of road conditions [96, 106] or traffic conditions [11]. Similar systems have been proposed to identify collision and accident events, and to collect digital information usable for forensics [148].

The third branch focuses on automatic detection and classification of users' transportation modality into a set of predefined transportation mode classes. Transportation mode detection systems typically rely heavily on machine learning methods, while physical motion sensing techniques are leveraged to produce descriptive features for the classification. In the rest of this section, we provide a concise survey on the transportation mode detection systems, focusing in particular on systems using motion sensing as the primary input source.

2.6.1 Transportation Mode Detection

Existing Transportation Mode Detection (TMD) research can be categorized using several criteria. For example Consolvo et al. [27] group related work based on the type of platform used, i.e., commercial devices, experimental research prototypes, and systems used to detect a specific physical activity. Reddy et al. [116] merge the first two into a single category and add a new category for mobile phones: commercial devices, custom hardware and mobile phones. In a more recent work Prelipcean et al. [111] emphasize the division of TMD literature between three research branches: Location-based services, Transportation science, and Human geography. Transportation mode detection can also be considered to be a subfield of Human Activity Recognition field (HAR). TMD work within HAR has roots in early work of the field where a subset of the identified activities were transportation-related modalities considered in TMD studies, e.g., walking, cycling or running [7]. The TMD within HAR field can be further split into work prior to independent transportation mode detection subfield, and work focusing solely on transportation mode detection. This split occurs roughly around year 2008 [132], before which transportation mode detection was rarely considered separately, and the transportation-related modalities were part of a broader set of considered activities.

From the application perspective, categorization can also reasonably be made based on the type of sensors utilized and consequently the type of data available for applications. For example, location data is often required by real-world transportation applications analyzing locations, POIs, and route trajectories. Systems collecting only low-level motion sensor data, on the other hand, allow self-contained systems regardless of the availability or stability of the signal environment, inherently also providing better privacy and security for the system. Finally, the impact of smartphones on transportation mode detection task has been so profound that a reasonable binary categorization could also be done based on whether a smartphone is used as the main platform or not.

In our review, we opt to use categorization based on platform and the main research branch of the work: (i) TMD within Transportation Sciences, (ii) TMD as a subfield of HAR with wearable and commercial devices, and (iii) TMD with smartphones. In the following sections, we review existing work on transportation mode detection, with emphasis on the studies that use motion sensing for the task.

TMD within Transportation Sciences

Transportation mode detection within transportation science primarily considers various techniques for offline interpretation of GPS trajectories. The first sensor-assisted, transportation-related work within the transportation science community considered augmenting electronic travel diaries with location measurements derived from a GPS sensor [32, 121]. These studies performed automatic spatiotemporal identification of trip start and end. We note that prior to May 1st 2000 the accuracy of civilian GPS localization was impaired by the US Selective Availability (SA) Act [46], which added artificial noise to GPS atom clocks in order to decrease accuracy of the localization. After termination of the SA and public availability of accurate GPS localization, the idea of automated travel diary was expanded by Wolf et al. [149] by addition of trip purpose and exact route taken between places, and by Ashbrook et al. [4] by clustering locations to identify significant places of the users, and establish the likelihood of travel between the discovered significant places. Stopher et al. [129, 130] investigate GPS-assisted travel surveys, while also touching on the topic of transportation mode detection. In their informal description of mode identification, the authors note that the use of speed in combination with GIS information on public transportation could be used to reliably identify the user's transportation mode.

One of the first works to specifically address identifying automotive transportation modalities was proposed by Patterson et al. [104], which leverages GPS measurements along with a GIS database detailing a bus transportation network. The authors model transportation network as a Graph and employ a particle filtering to identify between bus, car, and foot modalities. The work was later extended by Liao et al. [80] with an inclusion of a hierarchical activity model. Zheng et al. [156, 157] investigate extended features derived from GPS measurements to identify between stationarity, walking, biking, driving, and traveling by bus. The authors augment the standard GPS features, i.e., velocity and location, with segment-level features that capture variability in heading, velocity and acceleration. To identify segments, i.e., transportation periods defined

by a single continuous transportation mode, the authors identify walk segments as potential change points between modalities. In a closely related work, Zheng et al. [158] present GeoLife, a collaborative social networking service enabling interaction and recommendations between users based on interpreted location trajectories. As part of GeoLife, the authors present a transportation mode detection scheme based on GPS trajectories. The authors first employ a change point-based segmentation, yielding trip legs defined by a single modality. After obtaining trip legs, the authors employ an inference model similar to their previous work [156, 157]. Similarly, Xiao et al. [152] present a method for identifying transportation modes from GPS trajectories without the need for external GIS information. The authors capture a set of statistical features about speed, acceleration, azimuth between consecutive points, and sinuosity of the trajectory.

Stenneth et al. [127] leverage a combination of GPS and knowledge about the underlying transportation network, including real-time bus locations, to enable fine-grained classification of automotive modes. The authors revisit the GPS features proposed by Zheng et al. [157] opting to use average of heading change, acceleration and speed instead of frequency of exceeding a predefined threshold value. Bolbol et al. [13] explore limitations of the existing research, highlighting challenges in data collection, limited analysis to justify selected features, varying number of modalities considered, in particular for the automotive modes, and typical assumptions made to segment the data. The authors then propose a window-based SVM classification on GPS speed and GIS information on the locations of tube stations.

A number of studies investigating use of deep learning methods to infer transportation modes from location traces. For example Song et al. [125] propose a Multi-task Deep Long Short-Term Memory (LSTM)-based deep learning architecture to simultaneously infer transportation mode and predict users mobility. Endo et al. [33] and Wang et al. [137] investigate leveraging deep learning frameworks to automatically learn representative features from location trajectories for transportation mode estimation. From the results of the evaluation, the authors find that the features learned using the proposed approach can yield comparable or slightly improved performance when compared with classical hand-crafted features, demonstrating the feasibility of the method. In contrast to leveraging deep learning for feature learning, Dabiri and Heaslip [30] propose a CNN-based deep learning approach to directly predict transportation mode from location trajectories.

TMD within HAR using Wearable and Commercial Devices

Prior to the advent of electronic motion sensors, researchers utilized mechanical motion sensors to collect measurements about physical activities. Among the first such platforms were mechanical motion sensors, such as pedometers and actometers. These devices have been leveraged to estimate daily physical activity based on number of steps counted by the pedometer, and estimated energy expenditure from the actometer [92]. The introduction and miniaturization of electronic accelerometers allowed researchers to measure activities more precisely. For instance, Veltic and Bussmann [17, 135] employed multiple uniaxial accelerometers to investigate what the authors coined as the ambulatory accelerometry, detecting body posture, walking, and cycling activities at three different intensities.

Non-automotive activities related to transportation have been investigated in a plethora of studies using motion sensors employed either on a single sensing board or multiple wearable sensors. Examples of the former category include the *Sensing Badge* containing a bi-axial accelerometer [34], and the *eWatch* [65]. Using a custom sensing platform with embedded tri-axial accelerometer, Ravi et al. [114] demonstrated how selection of training and test data can drastically influence evaluation performance. A multi-sensor platform coined the *Mobile Sensing Platform* (MSP) was presented by Choudhury et al. [23]. The MSP, including the supported location sensing board, enables 9-DoF motion sensing through fusion of accelerometer, gyroscope and magnetometer. Additionally, the MSP contains an in-built activity recognition system able to differentiate between walking, running, cycling, using an elliptical trainer, and using a stair machine. Within the MSP sensing pipeline, the authors utilize boosting to automatically perform feature selection from a superset of 600 features.

Using multiple wearable accelerometers enables more detailed capture of human bio-mechanics, allowing differentiating between a more varied set of activities. For instance, Kern et al. [62] differentiate between 9 activities using 12 tri-axial accelerometers positioned at major joints on the human body. Bao and Intille [7] investigate a more varied set of activities with data collected by 20 users using five body-worn biaxial accelerometers. The authors consider a total of 20 activities, including transportation-related activities stationary, walking, running and cycling. In a study comparing a single sensor platform with multiple wearable sensors, Lester et al. [77] investigate practical activity recognition, paying attention to different sensor modalities used for activity recognition, number and placement of physical sensors worn, and cross-placement and cross-user generalization. Based on their results, the authors conclude that a single multisensor board is suffi-

cient for activity recognition with over 90% performance, with barometer, microphone and accelerometer deemed the most expressive sensor modalities. The authors note that training data should be collected from all of the expected sensor placements to cover placement-dependent variation of sensor measurements.

Troped et al. [132] consider combination of GPS and a triaxial accelerometer worn on the right hip to perform transportation mode detection between walking, running, cycling, skating, and car. The work was one of the first to specifically consider recognition of users transportation modality from the perspective of activity recognition domain[1]. In a similar system design, Feng and Timmermans [35] examine use of GPS and accelerometer sensors to determine users transportation mode as walking, running, bicycle, motorcycle, bus, car, tram, or metro, reporting a notably high, 91.7% overall recall of the detection. Instead of GPS, Sohn et al. [124] consider GSM-based transportation mode detection to discern between {stationary, walking, and automotive} activities. In a related system, Froehlich et al. [38] consider combination of accelerometer and GSM in a system coined as the *UbiGreen*. The transportation mode inference in Ubigreen combines Mobile Sensing Platform [23], providing automatic identification between sitting, standing, walking, running and cycling, with GSM-based vehicle identification inspired by the work of Sohn et al. [124].

After the surge of mobile phones with an ever-increasing array of embedded sensors, transportation mode detection with wearable devices has largely shifted into using smartphones as the primary platform. However, recent development of other sensor-equipped smart devices, such as watches, glasses and rings, has spurred a new wave of activity and transportation mode detection with (non-smartphone) wearable devices. In one such work, Bhattacharya and Lane [10] investigate the feasibility of deep-learning-based activity recognition on smartwatches. The authors sample the available sensors, i.e., accelerometer, gyroscope, barometer, magnetometer, light, temperature and WiFi, on the smartwatch model and apply a general-purpose Restricted Boltzmann Machine (RBM) to identify between four modalities: standing, walking, running, and motorized. The authors additionally consider a null-class, which corresponds to physical movement unrelated to transportation activities.

[1] we note however, that to best of our knowledge, the first work to consider recognition of automotive modalities using an accelerometer were Devaul and Dunn within the scope of the MIThril project [136].

TMD with Smartphones

A common technique for transportation mode detection on smartphones is to combine location information with motion sensing. For instance, Reddy et al. [116] study different combinations of sensors, features, and classification techniques to discern users' transportation mode between stationary, walking, biking, running and automotive. The authors find the fusion of accelerometer and GPS the most efficient sensor combination for the detection task, where GPS is used to measure instantaneous velocity, and accelerometer is used to capture users' overall motion characteristics. The work is among the first to consider using unconstrained smartphones for the task of transportation mode detection. Consequently, the authors use the orientation-agnostic L^2-norm as their motion representation. From the L^2-representation, the authors utilize correlation-based feature extraction, and find variance, and energy coefficients between 1-3Hz the best feature combination. Using a combination of C4.5 Decision Trees and Discrete Hidden Markov Model (DHMM), the authors employ a 10-fold cross validation to yield precision and recall of 93.7% and 93.6% respectively. We note that the study does not consider the distinction of fine-grained automotive modalities, which is generally recognized as one of the most challenging tasks for TMD systems, and which is considered within this thesis. Instead of GPS, Shin et al. [123] combine accelerometer with GSM to identify users' transportation mode between walk, bus, train/tram, and car. The authors use significant walking periods to segment transportation activities and GSM location to identify vehicular transportation.

Systems relying solely on motion sensors typically employ more elaborated motion sensing pipelines. For instance, Lu et al. [83] present a system called the *JigSaw* that includes a motion sensing pipeline for activity recognition, and evaluate its performance using transportation mode detection. The motion sensing pipeline consists of a preprocessing stage including virtual recalibration of the accelerometer, admission control to filter out extraneous activities, gravity component estimation based on a 4-second mean of the accelerometer measurements, and transforming the measurements into global vertical and horizontal motion representations. To alleviate the challenges related to unconstrained device placements, the authors propose a split-and-bound classification, where each transport modality is subdivided into placement-mode subclass, e.g., walk is divided into $pocket-walk$, $bag-walk$ and $jacket-walk$ classes. The authors consider five modalities: stationary, walking, running, cycling or vehicle, yielding an average accuracy of 95.1%. Wang et al. [143] investigate different motion representations in a system using mobile phone-embedded accelerometer to identify users'

transportation mode as stationary, walk, bike, car, bus, or subway. The authors produce three representations of the accelerometer measurements, i.e., L^2-norm, vertical linear acceleration and horizontal linear acceleration. As the gravity component estimation method, the authors apply average filtering with mean over an 8-second data frame. In the evaluation of their system, the authors find the best results with the L^2-norm (70.73%) compared to using the horizontal and vertical projections (60.71%). The under-performance of the projections is assumed to be caused largely by inaccuracies in the gravity estimation. We note that the added difficulty to discern between fine-grained automotive modalities is evident from the obtained detection performance between JigSaw and the system proposed by Wang et al. [143].

Other potentially effective sensors commonly found within mobile phones for transportation mode detection include barometer, magnetometer and microphone. Magnetometers have been shown to be effective in identifying rail-based transportation [140], whereas barometers can be used as a proxy for user's speed. As an example of a barometer-based approach, Sankaran et al. [118] demonstrate how barometers can be used to discern between coarse-grained modalities, i.e., idle, walking, and vehicle. Leveraging audio information sampled from a microphone for transportation mode detection is investigated by Wang and Roggen [142], where they utilize short time Fourier transform (STFT) and a CNN-based deep learning framework to detect between still, walk, run, bike, car, bus, train and subway. The authors find that the audio-based approach performs best, with roughly 95% precision, for distinguishing between fine-grained automotive modalities, which contrast and potentially complements the challenge of discerning between these modalities using motion sensing-based approaches. We note that when compared to motion sensors, the use of audio, magnetometers and barometers is more susceptible to overfit, as these sensor modalities are more prone to capture specific qualities related to routes and vehicles instead of the transportation mode in general.

Besides the above work specifically targeting TMD, several transportation sensing systems employ closely similar sensing pipelines. For instance, Wüstenberg et al. [88] differentiate between electric vehicles and fossil-fueled vehicles by inspecting the high-frequency components of motion. Nawaz and Mascolo [98] leverage mobile phone-embedded accelerometer and gyroscope to learn users' significant routes. In their sensing pipeline, sensor measurements are transformed into the global reference frame using gravity estimation based on low-pass filtering accelerometer measurements. The yaw rotation is then aligned with the vehicle's movement direction us-

ing a heuristic based on gyroscope-measured rotation during turning maneuvers. Similar coordination transformation from local device frame to vehicle-movement centered frame is performed by Mohan et al. [96] in their system coined *Nericell*. As part of the motion sensing pipeline, Nericell performs gravity estimation using a 10-second median filter to transform motion sensor measurements from the device's local to the global reference frame. The yaw rotation is resolved based on detecting braking and acceleration events from GPS, and estimating the direction of acceleration within the accelerometer measurements during these events.

In an effort to provide a benchmark dataset for the transportation mode detection community, Gjoreski et al. [43] have collected and annotated 2812 hours of multi-sensor data about transportation activities within south-east UK. The dataset, coined as the Sussex-Huawei Locomotion-Transportation (SHL) dataset, was collected using a single device model carried by 3 users in 4 different placements over a period of 7 months. The dataset contains measurements from 16 different sensor modalities available in the target device. The measurements are manually annotated with details about 8 transportation modes, participants posture, inside/outside location, road conditions, presence of a tunnel, social interactions and having meals. A subset of the SHL dataset was employed by Wang et al. [139] in Sussex-Huawei Locomotion Challenge[2]. The challenge involves participants with a competition to recognize eight transportation modes: still, walk, run, bike, car, bus, train, and subway using measurements from four hardware sensors, i.e., accelerometer, gyroscope, magnetometer and barometer, and three virtual sensors, i.e., linear acceleration, gravity and orientation sensors. The measurements are collected from pocket placement from one user over a period of 82 days. Of the 19 submissions, 11 leveraged classical machine learning pipeline (ML), and 9 deep learning (DL) frameworks. On average the DL approaches performed marginally better than the ML ones, with the best DL system yielding an average F_1-score of 93.9% and the best ML system 92.4%. The best system utilizing only inertial sensors, i.e., accelerometer and gyroscope yielded an average F_1 score of 88.8%, ranking 3rd in the competition. The system, called *Tesaguri*, uses a CNN-based deep learning approach with input as the spectra of raw sensor measurements. Interestingly, a benchmark system designed by Wang et al. [139], based similarly on frequency transformation of raw sensor measurements (accelerometer, gyroscope and magnetometer) and CNN-based deep learning approach yields an overall F_1-score of 92.9%, further implicating suitability of CNN for the transportation mode detection task.

[2] `http://www.shl-dataset.org/activity-recognition-challenge/`

Chapter 3

Linear Acceleration for Transportation Sensing

Accurately capturing periods of *linear acceleration* is one of the main challenges for motion sensing on mobile devices. While a powerful representation of motion information directly related to the true movement of the user, linear acceleration is easily masked by acceleration caused by gravity, and distorted by noise and imperfect sensing equipment. When transformed into global reference frame, linear acceleration enables orientation-agnostic activity recognition [83], characterizing vehicular movement [42, 52, 58], augmenting positioning systems through inertial navigation [150] and assessing risk of injury [87]. However, accurately performing such transformation is difficult during periods of sustained linear acceleration, when linear acceleration is mixed with acceleration due to gravity, and thus making estimation of the gravity component difficult.

Within transportation, sustained linear acceleration on the horizontal plane is the main resultant force measurable by motion sensors from substantial changes in velocity or change in the direction of movement. The ability to capture and characterize these periods is a key enabler for sensing applications targeting mechanized and automotive transportation. In this chapter, we present novel approaches to capture sustained linear acceleration for both accelerometer-based and IMU-based motion sensing. Using these techniques, we propose a novel set of *peak* and *segment* features, usable for both physical sensing and machine learning systems to monitor users' transportation behavior. Finally, based on the presented techniques, we describe an accelerometer-based transportation mode detection system capable of discerning between fine-grained automotive transportation modalities. The contributions presented in this chapter are based on Article I and Article II.

Algorithm 1 $G_{est}^{(x,y,z)} = \texttt{Gravity}\ (A_i^{(x,y,z)})$

$F_{mean}^{(x,y,z)} = mean(A_i^{(x,y,z)})$
$F_{var}^{(x,y,z)} = var(A_i^{(x,y,z)})$
for each $a \in \{x, y, z\}$ **do**
    if $F_{var}^{a} \geq \tau_{varMax}$ **then** ▷ currently we use $\tau_{varMax} = 1.5$
        return $G_{est}^{(x,y,z)} = \texttt{AverageFilter(5s)}$
    end if
    if $||F_{mean}^{a} - G_{est}^{a}|| \geq \tau_{\Delta G}$ **then** ▷ currently we use $\tau_{\Delta G} = 2m/s^2$
        $\tau_{var}^{a} = F_{var}^{a} + \epsilon$ ▷ Force update of the gravity component
    end if
    if $F_{var}^{a} < \tau_{var}^{a}$ **then**
        $G_{est}^{a} = F_{mean}^{a}$
        $\tau_{var}^{a} = (F_{var}^{a} + \tau_{var}^{a})/2$
        $Increase^{a} = \tau_{var} * \epsilon_{inc}$
    else
        $\tau_{var}^{a} = \tau_{var}^{a} + Increase^{a}$
    end if
end for

3.1 Adaptive Accelerometer-based Gravity Estimation

Ideally, the gravity component could be calculated directly from accelerometer measurements, allowing self-contained and low-energy systems to track movement within global frame regardless of device orientation. However, isolating the gravity component from other sources of acceleration, particularly within the low-grade accelerometer sensors commonly embedded in modern mobile devices, is inherently a heuristic process, providing only an estimate of the gravity component. Our approach for accelerometer-based gravity component estimation, *adaptive accelerometer-based gravity estimation* (*AAGE*), can be understood as an extension of the opportunistic gravity estimation method described in Section 2.4.3. Our approach, detailed in Algorithm 1, is designed to provide more robust and accurate gravity component estimation during mechanized transportation, alleviating the limitations outlined in Section 2.4.3. Specifically, in our approach we propose:

(1) Adapting the variance threshold τ_{var}, used to control triggering gravity estimates. This ensures finding a suitable threshold value respective of the underlying motion and noise environment. To ensure estimates are obtained, the variance threshold is relaxed over time, proportionally to the current threshold, until a gravity estimate is triggered, or variance of any of the accelerometer axes exceeds the upper limit of $\tau_{varMax} = 1.5$ (line 4). When a gravity estimate is triggered, i.e., when variance of the current frame F_{var} is below τ_{var}, we set the current gravity estimate G_{est} to the mean of the current 1-second frame F_{mean}, and set the variance threshold τ_{var} to the average of the current threshold value and variance of the current data frame F_{var} (lines 11 – 12). The combination of relaxing and tightening of the variance threshold value aims to ensure that it remains suitable for the current motion and noise environment. Additionally, this procedure provides a mechanism to control the balance between frequency and quality of the gravity estimation points. Finally, opportunistic gravity estimation remains a valid approach only when the overall variation of acceleration is low. Consequently, when accelerometer variance F_{var} along any of the sensor axes increases above a predefined threshold $\tau_{varMax} = 1.5$, typically produced during pedestrian activities, we connect our method with average filtering using a 5-second frame length (lines 4 – 5).

(2) Detecting notable shifts in device inclination, and reinitializing gravity estimation in these cases. Specifically, we monitor the difference between the mean of 1-second frame F_{mean} and current gravity estimate G_{est}. Whenever the difference exceeds a predefined threshold $\tau_{\Delta G} = 2m/s^2$, the threshold τ_{var} is set to $F_{var} + \epsilon$ (lines 7–8), where ϵ is a small constant, forcing an update of the gravity estimate update (lines 10–14). This procedure aims to mitigate the detrimental effects of large inclination shifts to the gravity component estimation. Note that the selection of $\tau_{\Delta G}$ defines the maximum sustained linear acceleration detectable by the algorithm, effectively creating a trade-off between maximum sustained linear acceleration, and maximum error producible by shifts in device inclination.

To validate the effectiveness of the AAGE, we have performed a small-scale experiment comparing our gravity estimation technique with accelerometer filtering. As the filtering technique, we consider mean over data frame using 1, 10 and 30 second frame duration. Our experiment dataset contains slightly over 7 hours of accelerometer data collected from four different automotive transportation modalities. As the evaluation metric, we compare

	Our	Miz-1	Miz-10	Miz-30
Bus	0.32	0.09	0.24	0.27
Train	0.54	0.16	0.25	0.42
Metro	0.51	0.18	0.32	0.41
Tram	0.35	0.09	0.31	0.32

Table 3.1: Correlation coefficient between integral of the horizontal linear acceleration and GPS speed sampled at 1Hz. As the gravity estimation methods, we consider our method and average filtering approach with three typical data frame lengths.

velocity obtained through numerical integration of linear horizontal acceleration with velocity obtained from GPS measurements sampled at 1Hz frequency. The results of this evaluation are presented in Table 3.1. From the results, we observe that our approach yields consistently higher correlation coefficient with GPS velocity when compared to average filtering using different data frame lengths. Another key benefit of our approach is that it relies on short data frame duration, allowing rapidly reacting to changes in the gravity component, and thus reducing the risk of large errors due to latency in the gravity component estimation.

3.2 IMU-based Adaptive Gravity Estimation

As detailed in Section 2.4.3, addition of a gyroscope enables realizing the device as a 6-DoF IMU, allowing more robust separation between periods of linear acceleration and shifts in device inclination. To take advantage of the added motion information provided by IMU, we have designed an extended version of AAGE, referred to as *IMU-based Adaptive Gravity Estimation* (*IAGE*). IAGE is based on four modules, each handling a specific task for gravity component estimation:

1. Keypoint Detection and Validation,
2. Orientation Change Detection,
3. Gravity Estimation Interpolation, and
4. Linear Acceleration Decomposition

The first two modules (1) – (2) correspond to AAGE's operating principle, i.e., (1) opportunistically identifying periods when the main force

affecting the device is gravity, and (2) separately detecting and handling large orientation shifts in the device inclination. Within the scope of IAGE, we refer to the near-static periods as *keypoints*, and the module responsible for finding these periods as *Keypoint Detection and Validation*. The module responsible for detecting and handling large shifts in device inclination is referred to as *Orientation Change Detection*. The two remaining modules (3) – (4) are unique for IAGE, and are designed to take advantage of the measurements about device rotation. The first of the new modules (3) *Gravity Estimation Interpolation*, is responsible for interpolating the gravity component between the keypoints. The final module (4) *Linear Acceleration Decomposition* separates tangential and radial components of the horizontal, gravity-eliminated linear acceleration during mechanized movement. In the following subsections we provide an overview of each of these modules.

3.2.1 Keypoint Detection and Validation

To identify optimal keypoints for gravity estimation we follow an opportunistic approach, similar to AAGE, where the threshold to trigger gravity estimation is adapted over time to match the current motion environment. As the main difference, instead of simple accelerometer variance, we measure *stability* S of the motion environment using a combination of acceleration and rotation measurements. Stability reflects the overall motion variability within a data frame. Formally, stability of frame x is estimated as:

$$S(i) = \alpha_\sigma \sigma_i + \alpha_\mu \delta\mu_i + \alpha_\omega |\omega_i|, \tag{3.1}$$

where σ_i denotes the standard deviation of the accelerometer measurements for frame i, $\delta\mu_i$ is the difference in accelerometer mean between subsequent data frames i and $i-1$, and $|\omega_i|$ is the magnitude of angular velocity for data frame i. The difference in mean provides a mechanism for detecting sustained acceleration, whereas angular velocity enables accounting for periods where the device is influenced by centripetal force. The variables α_σ, α_μ, and α_ω are weight terms, which we require to sum up to one. In the experiments we consider $\alpha_\sigma = 0.1$, $\alpha_\mu = 0.45$ and $\alpha_\omega = 0.45$ as the values of the weighting coefficients. We note that the distribution of α values does not directly reflect the importance of the different components, as the units of the three components are not in-scale. In practice, the selected α values allocate a roughly even weight to the different components. Finally, we

Algorithm 2 `FindKeypoints` (Ω_i, A_i)

1: $S_i = Stability(\Omega_i, A_i)$
2: **if** $S_i \leq \tau_s$ **then**
3: $\quad K_{stab} = S_i$
4: $\quad K_{grav} = mean(A_i)$
5: $\quad keyPoints.add(K)$
6: $\quad \tau_s = S_i$
7: $\quad Increase = S_i * \epsilon$
8: **else**
9: $\quad \tau_s = \tau_s + Increase$
10: **end if**

note that the stability score measures overall variability of the measurement frame, and hence a low score implies high stability for a keypoint.

After calculating the stability score, the keypoint detection algorithm, presented in Algorithm 2, follows a similar pattern to AAGE of adapting the threshold τ_s to generate new keypoints, effectively balancing between the frequency and quality of generated keypoints. Specifically, we compare the stability score S_i (Eq. 3.1) of the current data frame i against an adaptive threshold τ_s (Lines 1-2). Whenever the stability of current frame is within the threshold, we create a new keypoint K that consists of a stability score K_{stab} and a gravity estimate K_{grav}. As the stability of the keypoint K_{stab} we use the stability of the corresponding data frame i (Line 3). As the gravity estimate K_{grav} we assign the mean of the accelerometer measurements of the data frame i (Line 4). When the stability score S_i of the current frame i exceeds threshold τ_s, we increase τ_s by a value relative to the stability of the most recent keypoint K_{stab} using a small scalar ϵ (Line 9). In the experiments we use $\epsilon = 0.01$. The threshold is increased until the stability of a new frame falls within the threshold, after which the threshold τ_s and the increase value are reinitialized using the stability of the current frame (lines 6–7). Thus, the algorithm effectively searches for the lowest possible score value for the current inertial environment, while at the same time balancing between the recentness of the generated keypoints. For an illustration of the gravity estimation, see Figure 3.1.

While the keypoint generation algorithm generally performs well, stability can remain low during periods of constant sustained linear acceleration, triggering generation of *false keypoints*. Such false keypoints are problematic for two reasons. First, they result in loss of motion information

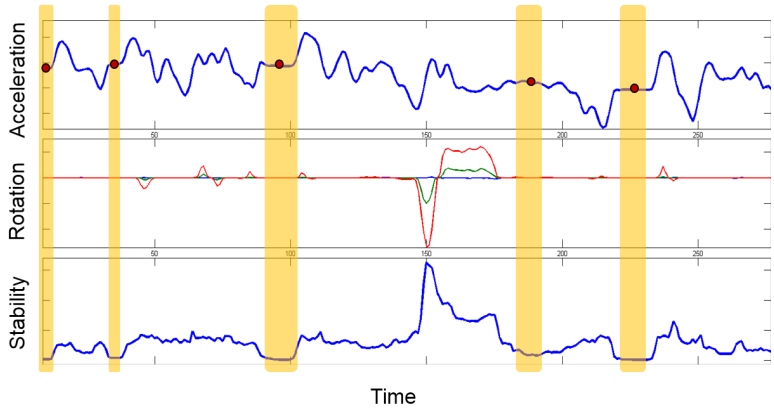


Figure 3.1: Illustration of the keypoint generation process within IAGE. The red dots correspond to generated keypoints K.

about the sustained linear acceleration event triggering the false keypoint. Second, the false keypoint can produce erroneous gravity estimates while estimating gravity between keypoints, resulting in false linear acceleration. To reduce the risk of producing false keypoints, we perform a separate keypoint validation step on the stored keypoints before using them for gravity estimation. The intuition within the keypoint validation is that whenever accelerometer measurements report a change in device inclination $\Delta\Theta_a$, a change of approximately same magnitude $\Delta\Theta_r$ should be measured by the gyroscope; see Figure 3.2 for illustration. In practice, limited sensor quality and the complexity of the user's inertial environment always cause a level of disagreement with the two inclination estimates. To account for these factors, we define an error range ϵ_Θ, which defines the acceptable margin of error between the two inclination changes. If the threshold is exceeded, the keypoint is considered unreliable and filtered from the final gravity estimate.

3.2.2 Orientation Change Detection

Efficiently handling large and sudden changes in device inclination is one of the key challenges in gravity component estimation. If not treated properly, these events can result in large errors to the gravity component while the algorithm converges to the new device orientation. Within the scope of AAGA, we presented an orientation change detection method relying on

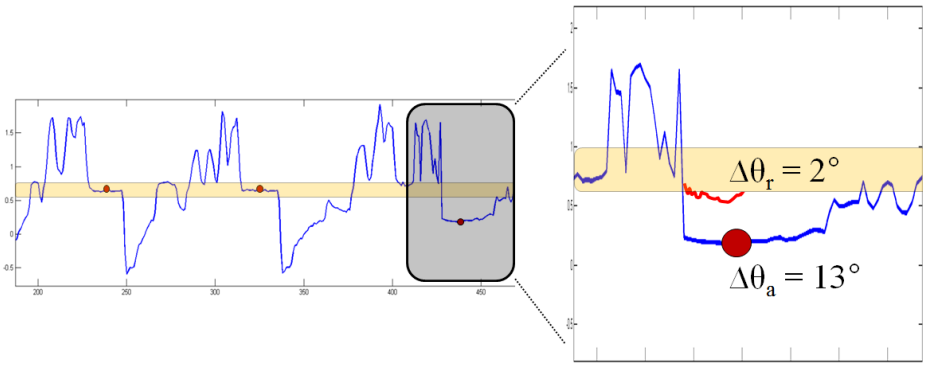


Figure 3.2: Illustration of the procedure to detect false Keypoints generated during an event of near-constant sustained linear acceleration. The blue line notes accelerometer measurements over time and the red line notes inclination tracked by integrating gyroscope measurements. The $\Delta\Theta_r$ and $\Delta\Theta_a$ note the change in device orientation indicated by integration of the measured rotation and accelerometer respectively.

monitoring large changes between current gravity estimate and accelerometer measurements. This approach, however, has a limitation that it is unable to discern periods of strong linear acceleration that exceed a predefined threshold from actual changes in device inclination. Within the scope of IMU-based systems, the inclusion of a gyroscope enables addressing this limitation by directly measuring changes in device orientation.

Our IAGA orientation change detection algorithm, presented in Algorithm 3, follows a two-stage approach to *detect* and *validate* problematic shifts in device inclination. In the first stage, we detect large shifts in device orientation relative to prevalent motion environment. Specifically, to ensure our algorithm adapts to current motion environment, we utilize a stream-based event detection approach to detect potentially problematic orientation changes. The event detection algorithm compares current change in gyroscope magnitude against a threshold τ_Ω that is determined dynamically based on a running estimate of the standard deviation of gyroscope magnitude (Lines 1, 12). Currently we set the threshold τ_Ω equal to three times the standard deviation of recent gyroscope measurements (Line 12), which approximates a two-sided t-test with a significance value of 0.0001.

Algorithm 3 `OrientationChange`

1: $\Omega_{L^2} = getMagnitude(\Omega)$
2: **if** $\Omega_{L^2} > \tau_\Omega$ **then** ▷ Detect
3: $\Theta_i = getEulerInclination(G)$
4: $\Theta_{i-n} = getEulerInclination(G_{i-n})$
5: $\Delta\Theta = abs(\Theta_i - \Theta_{i-n})$
6: **if** $\Delta\Theta > \tau_\Theta$ **then** ▷ Validate
7: **do** ▷ Orientation change confirmed
8: $HighKinematicTracking()$
9: **while** $\Omega_{L^2} > \tau_\Omega$
10: **end if**
11: **else**
12: $\tau_\Omega = 3\sigma(\Omega_{L^2}^{1..i})$
13: **end if**

In the second stage of the algorithm we validate that the inclination of the device has indeed changed. This step is relevant to avoid triggering the algorithm unnecessarily, as large angular velocities do not necessarily translate to shifts in device orientation. For instance, rotations around the gravity vector do not influence the device inclination. Additionally, a bump in the road or the user shifting position can produce large momentary angular velocities. However, the movement in these situations is often symmetric in the sense that the device returns to approximately initial orientation, thus resulting only in small changes to device inclination and keeping the existing gravity estimate valid. The resulting change in device inclination from an event of high angular velocity is calculated by comparing the initial inclination (Line 3) with the inclination measured at the end of the high angular velocity event (Lines 4 - 6). When the change in inclination exceeds a predefined threshold $\tau_\Theta = 10°$ (Line 6), we assume the inclination of the device has changed substantially and connect our method with *high-kinematic orientation tracking*, which ensures that the inclination change does not produce false linear acceleration. As the high-kinematic orientation tracking technique we employ the method of Madgwick [84]. Once orientation no longer changes, we store the current orientation and reset the gravity estimate.

3.2.3 Gravity Interpolation

As the final step for IAGA, we interpolate the gravity component between the generated keypoints. The interpolation is handled by two algorithms:

propagation which provides real-time gravity estimates updates from the most recent keypoint K_a, and *filtering* which is triggered once a new keypoint K_b is detected and enhances the gravity component estimation between K_a and K_b.

Propagation updates gravity component estimates from the most recent keypoint K_a using a complementary filter [25] that balances between accelerometer and gyroscope measurements, formally

$$G(x) = (1 - \alpha) * G(x - 1) * \boldsymbol{R}_\Omega + \alpha * A(x)_L^{(x,y,z)}, \tag{3.2}$$

where α is a weight coefficient and $\boldsymbol{R}_\Omega$ is the rotation matrix derived from gyroscope measurements

$$\boldsymbol{R}_{\boldsymbol{\Omega}} = \begin{bmatrix} 1 & -\Omega_z & \Omega_y \\ \Omega_z & 1 & -\Omega_x \\ -\Omega_y & \Omega_x & 1 \end{bmatrix}. \tag{3.3}$$

The value of α, i.e., the weight allocated to the accelerometer in order to negate gyroscope drift, is determined adaptively based on the relative frame stability:

$$\alpha = 0.01 * \frac{K_a^S}{Stability(X_i)}. \tag{3.4}$$

This approach aims to allow the interpolation to rapidly correct drift when the device is in a stable state, while at the same time reducing loss of motion information during periods of linear acceleration. The value 0.01 reflects a commonly-used α-value in complementary filters, and is used as a base value for the α-adaptation.

Filtering takes advantage of a newly generated keypoint K_b by performing a bi-directional gravity interpolation. Specifically, we compute two gravity component estimates using forward interpolation starting from K_a: $G_a = G(x_{1\ldots n})$, and backward interpolation starting from K_b: $G_b = G(x_{n\ldots 1})$. For both interpolations, the weight term α used in the complementary filter is adapted to use $\min(K_a^S, K_b^S)$ as the denominator. The reliability of both gravity interpolations can be assessed by comparing the interpolation error E between the final gravity component value with the corresponding keypoint: $E_a = \|G_a(x_n) - K_b^G\|$ and $E_b = \|G_b(x_1) - K_a^G\|$. The fused gravity estimate is then calculated as a weighted combination

of the interpolations G_a, G_b, with weights determined by the interpolation error E. As the weight function, we use a non-linear Sigmoid function:

$$\begin{cases} W = 0 \ , i = 1 \\ W = 2(\frac{x}{n})^2 \ , 1 < i < c \\ W = 1 - 2(\frac{x-n}{n})^2 \ , c < i < n \\ W = 1, i = n \end{cases} \tag{3.5}$$

where n is the number of measurements between keypoints and $c = \frac{nE_a}{E_a+E_b}$ is a weighting factor determining the relative interpolation quality between the forward and backward estimates. The final interpolation G can now be expressed as:

$$G = (1 - W)G_a(x_{1...n}) + WG_b(x_{n...1}). \tag{3.6}$$

3.2.4 Linear Acceleration Decomposition

Recall from Sections 2.2.1, 2.4.4 that, using the estimated gravity component G, linear acceleration $\acute{a}$ within the device's local frame can be estimated by subtracting the gravity component from accelerometer measurements. Next, the linear acceleration can be rotated to global reference frame respective to inclination by using the gravity component as a reference.

The (directional) vertical component of linear acceleration within the global reference frame can then be directly read from the z-axis of the rotated, gravity-eliminated accelerometer measurements $\acute{A}^z_{G(\Theta)}$. The horizontal component of linear acceleration $\acute{a}^h_{G(\Theta)}$ corresponding to $\acute{A}^{(x,y)}_{G(\Theta)}$ is more difficult to estimate as there is no direct reference point between the global and local frames within the horizontal plane. As a partial solution, we propose a heuristic approach during mechanized transportation, which is able to decompose the horizontal linear acceleration into tangential and radial components respective to the movement direction of the device

$$h = h(t) \cdot h(r). \tag{3.7}$$

Our solution relies on identifying and leveraging common mechanized transportation maneuvers corresponding to *accelerating*, *braking*, and *turning*. The three maneuvers are identified by comparing correlation ρ between the horizontal linear acceleration $\acute{A}^h_{G(\Theta)}$ and rotation around the gravity component, corresponding to a change in yaw $\Delta\psi$, given by the z-axis of

the rotated gyroscope measurements $\Omega^z_{G(\Theta)}$. The presence of strong correlation indicates a turning maneuver whereas the lack of yaw rotation during linear acceleration indicates an accelerating or braking event.

During the first two maneuvers, i.e., acceleration and braking, we rely on an assumption that the majority of the horizontal linear acceleration is caused by tangential acceleration due to velocity change. During these periods, the direction angle of tangential horizontal acceleration can be estimated using a four-quadrant inverse tangent:

$$\alpha_V^{h(t)} = \arctan(\acute{A}^x_{G(\Theta)}, \acute{A}^y_{G(\Theta)}), \tag{3.8}$$

where V denotes the vehicle-centered reference frame. Similarly, we assume that turning maneuvers will result in centripetal acceleration, revealing the radial component of horizontal linear acceleration. We use correlation between horizontal linear acceleration and rotation around the gravity component as weighting terms

$$\rho_x = corr(R^z_{G(\Theta)}, \acute{A}^x_{G(\Theta)}) \tag{3.9}$$

$$\rho_y = corr(R^z_{G(\Theta)}, \acute{A}^y_{G(\Theta)}) \tag{3.10}$$

and estimate the direction angle of the radial direction respective to the vehicle-centered frame using:

$$\alpha_V^{h(r)} = \arctan(\frac{\rho_x \acute{A}^x_{G(\Theta)}}{||\rho||}, \frac{\rho_y \acute{A}^y_{G(\Theta)}}{||\rho||}). \tag{3.11}$$

3.2.5 Evaluation of Linear Acceleration

As one of the evaluations of IAGE, we compare accumulation of error between the ground truth distance obtained from GPS measurements, sampled at 1Hz, and the distance obtained through double integration of tangential horizontal acceleration $\acute{a}_V^{h(t)}$. The evaluation is performed by first partitioning the ground truth data into segments using zero-velocity stopping periods as deliminator. Each segment then corresponds to transportation between two consecutive stops. We use the segment-based evaluation to provide a measure of control for the accumulation of drift, in order to avoid propagation of large errors between segments and masking the actual performance of the algorithms. We compare our method with average

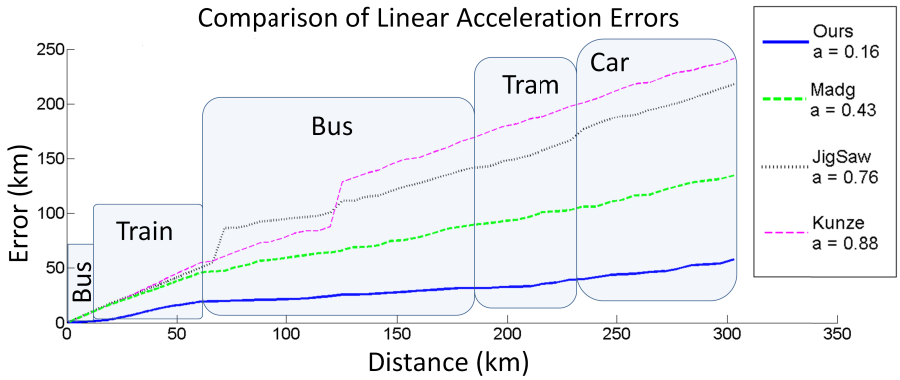


Figure 3.3: Comparison of the accumulation of distance error of the evaluated approaches, where a notes regression coefficient. The comparison is performed against a distance obtained from GPS measurements.

filtering used in JigSaw, proposed by Lu et al. [83], opportunistic gravity estimation proposed by Kunze et al. [67], and the IMU-based solution proposed by Madgwick et al. [84]. The results of the evaluation are illustrated in Figure 3.3, demonstrating that our method is able to track actual distance during mechanized transportation within approximately 16% accuracy. Compared to the considered baselines, our approach is able to significantly improve the correspondence with the ground truth, ranging from a 27% increase over the IMU-based Madgwick baseline, to a 71% increase over the Kunze baseline. For further evaluations and technical details, we refer to the attached Article II.

3.3 Feature engineering for Linear Acceleration

Recall from Section 2.5.2 that motion information is typically transformed into features for machine learning inference. Features are typically computed from data frames of fixed duration, and are thus referred to as *frame-based* features. While the standard frame-based features can effectively characterize transient and high-frequency motion common for human activities, they are ill suited for capturing periods of sustained acceleration. This limitation is particularly evident during mechanized transportation, where the most significant motion information is produced by acceleration, deceleration and turning maneuvers. Within the motion sensor measurements, these periods are evident as sustained elevation (or declination) *peaks* within linear horizontal acceleration and rotations around the grav-

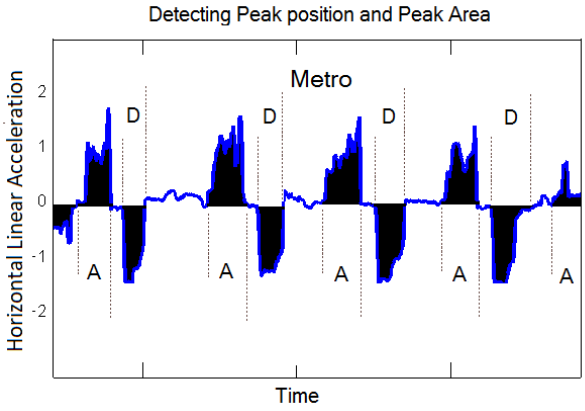


Figure 3.4: Peak areas detected from gravity-eliminated horizontal acceleration during a metro ride.

ity vector; see Figure 3.4 for illustration. To quantify these peaks in a way that is usable for machine learning methods, we design a collection of features, called the *peak features*.

3.3.1 Peak Features

Peak features are statistical features that quantify identified peaks within the input signal. As the first step of extracting peak features, we identify *peak areas* corresponding to a significant elevations within the monitored motion representation. We trigger a peak using a stream-based event detection algorithm, based on significant change in the monitored motion representation. After triggering a peak, we mark the start time of the event and buffer subsequent measurements until the measurements even out. Once the peak area is defined, we calculate a set of features quantifying key aspects of the peak: peak length, maximum value, volume, excess kurtosis and skewness; see Figure 3.5 for an illustration of the different features. An additional benefit of peak features is that they are able to maintain an intuitively clear relationship with the underlying physical phenomena. Specifically, the first two features captures duration and intensity of the peak, while peak volume capture total velocity change or heading change, and kurtosis and skewness capture symmetrical properties of the peak event.

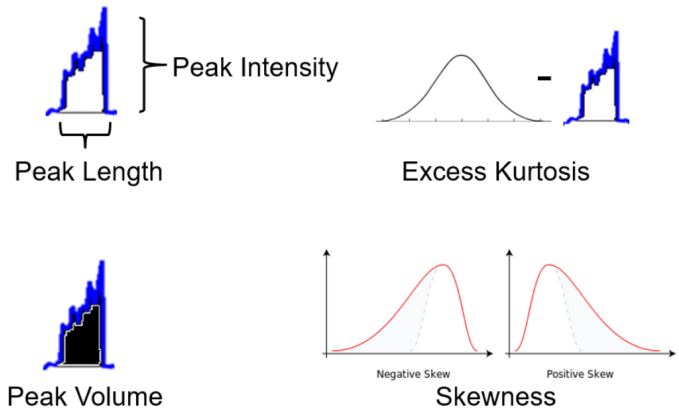


Figure 3.5: Peak features are a collection of features quantifying the key metrics of the peak event.

3.3.2 Segment Features

In addition to the peak features, we propose a set of *segment features* that capture motion patterns over a longer time span, e.g., over a single trip with automotive transportation. On a conceptual level, segment features correspond to the segment-scale features proposed by Zheng et al. [157], i.e., heading change rate, velocity change rate and stop rate. In addition to monitoring the frequency of the peak events, we propose monitoring frequency, mean duration and variability of the duration of the intermittent stationary periods, and variability of the peak features.

3.4 Transportation Mode Detection

To demonstrate the potential of the proposed peak and segments features, we have developed the accelerometer-based transportation mode detection system described in Article I. The underlying intuition for using peak- and segment-based features for transportation mode detection task is illustrated in Figure 3.6, depicting horizontal linear acceleration for different automotive modalities. In the figure, we can observe that the motion patterns revealed by horizontal linear acceleration are similar within the same modality, while distinct from other modes, providing a strong intuitive basis for our approach.

The performance of smartphone-based activity recognition approaches is typically sensitive to the placement of the device. Peak- and segment-

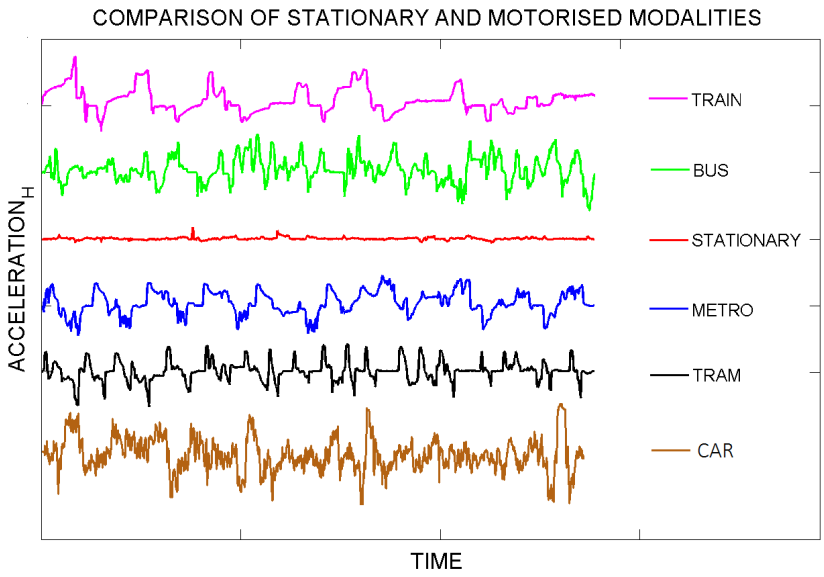


Figure 3.6: Comparison of the horizontal acceleration profiles for different motorized transportation modalities.

based features, however, characterize movement patterns of vehicles instead of humans, making these features more robust against variation over different sensor placements. To demonstrate this aspect, consider Figure 3.7 presenting horizontal linear acceleration during a single tram ride from three different users and three different phone placements: bag (top, magenta), trouser pocket (middle, red) and jacket (bottom, blue). In the figure, we can observe that the horizontal acceleration profiles are nearly identical, demonstrating that the peak- and segment-based features are less susceptible to variation due to device placement.

3.4.1 Sensing pipeline

Our sensing pipeline follows a standard motion-sensing architecture composed of a preprocessing, gravity estimation and feature extraction modules. As the first step, we buffer accelerometer measurement into short data frames of 1.2-second length, ensuring that our system is able to react rapidly to changes in transportation mode. We preprocess the raw sensor measurements using a low-pass filter which retains 90% of the original signals energy. Using a dynamic threshold based on the current motion intensity is designed to preserve more of the high-frequency components during

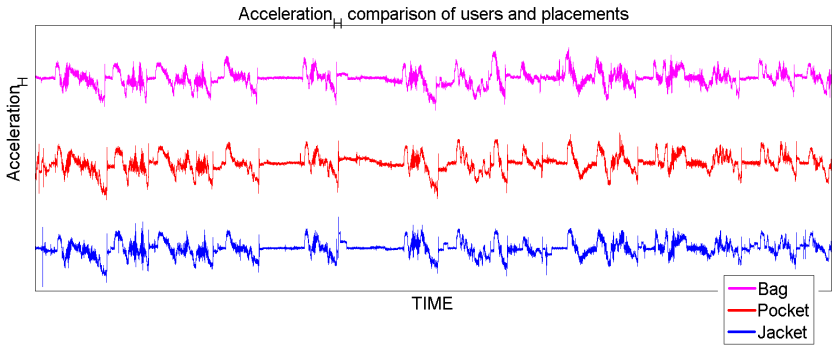


Figure 3.7: Unfiltered horizontal acceleration during a tram ride for three different users, with phones placed in three different placements.

mechanized transportation, while simultaneously allowing more substantial filtering during pedestrian modalities.

Next, we utilize the AAGE approach for gravity estimation, described in Section 3.1, to approximate the gravity component within the accelerometer measurements. Using the gravity component estimation, motion information is then transformed onto a global reference frame respective to device inclination. Since our system does not require a gyroscope, it is unable to leverage measurements about angular velocity to reliably distinguish between radial and tangential components of horizontal linear acceleration. Instead, as an alternative heuristic solution, we perform PCA on the two-dimensional horizontal linear acceleration and project the horizontal linear acceleration along the resolved principal components. We then sum the two resulting, orthogonal vectors into a single-dimensional representation of overall horizontal linear acceleration. Compared to simply using L^2-norm of horizontal linear acceleration, the advantage of this method is that it allows retaining the direction of linear horizontal acceleration, allowing a separation between acceleration and deceleration periods of mechanized movement. Finally, since we assume arbitrary device orientation, the sign of the linear horizontal acceleration does not directly correspond to increase or decrease of velocity. As a simple heuristic, we use the intermittent stationary periods to resolve sign of the linear acceleration by assuming that acceleration before a stationary period corresponds to decrease in velocity, whereas sustained linear acceleration after a stationary period corresponds to increase in velocity. As the final motion representations, we consider the L^2-norm of the overall acceleration, vertical linear acceleration $\acute{a}^v_{G(\Theta)}$, and horizontal linear acceleration $\acute{a}^h_{G(\Theta)}$.

Representation	Features
Statistical	Mean, STD, Variance, Median, Min, Max, Range, Interquartile range, Kurtosis, Skewness, RMS
Time	Integral, Double integral, Auto-Correlation, Mean-Crossing Rate
Frequency	FFT DC component, Spectral coefficients at 1,2,3,4,5,6 Hz, Spectral Energy, Spectral Entropy, Spectrum peak position, Wavelet Entropy, Wavelet Magnitude
Peak	Volume (AuC), Intensity, Length, Kurtosis, Skewness
Segment	Variance of peak features (10 features), Peak frequency (2 features), Stationary duration, Stationary frequency

Table 3.2: Full list of the features considered for the transportation mode classifiers.

From each motion representation, we extract *frame based*, *peak based*, and *segment based* features. The frame-based features are selected based on feature analysis presented by Figo et al. [36], while the peak- and segment-based features correspond to the features presented above in Section 3.3. The full list of considered features is presented in Table 3.2.

3.4.2 Machine learning inference

To identify the user's transportation mode, we use a three-stage hierarchical classification framework that decomposes the transportation mode detection task into specific subtasks. The underlying idea, illustrated in Figure 3.8, is to proceed from coarse-grained transportation mode classification towards a more fine-grained distinction of the current modality. The advantage of this approach is the ability to provide information at different granularity and confidence, depending on the application requirements.

To improve the performance of our transportation mode detection system, we rely on a logical assumption that consecutive automotive modalities are separated by intermittent segments of pedestrian activity [156]. To leverage this observation, we perform segment-wise classification of non-pedestrian transportation modes instead of solely relying on frame-by-frame classification. Specifically, as more evidence to support one of the

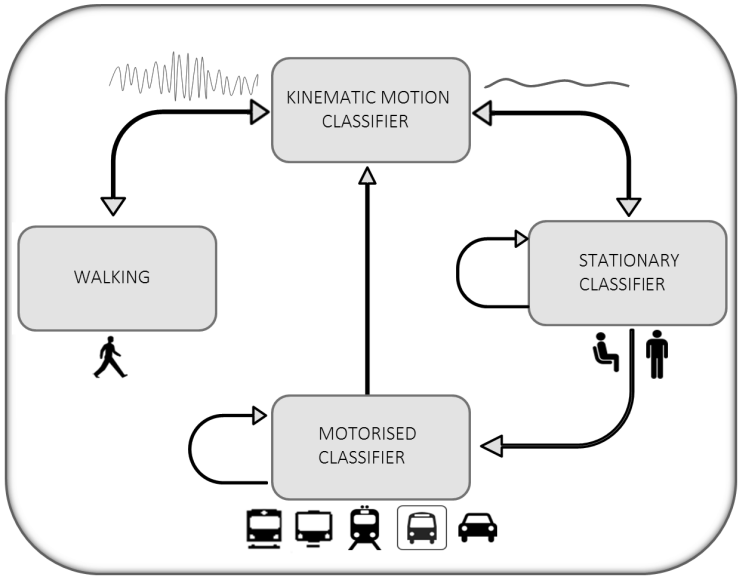


Figure 3.8: Overview of the classifiers used in our system and their dependencies.

modalities accumulates during a segment, our system is able to provide an increasingly accurate prediction of the current transportation mode. We continue the segment-wise classification until a pedestrian activity is detected. We note that stationary periods during automotive transportation are interpreted as being *stopped* in a vehicle, and are treated separately from stationary periods outside automotive transportation.

Kinematic Motion Classifier

At the root of the hierarchical classification scheme is a *kinematic motion classifier* which differentiates high-kinematic pedestrian modalities from the other, low-kinematic modalities. As the kinematic motion classifier scheme, we use a well-established technique of combining an instance-based classifier with a generative classifier [116]. As the instance-based classifier we utilize an ensemble of decision trees (with $T = 10$ trees) of depth one, trained using Adaptive Boosting [37]. The ensemble classifier leverages frame-based features, where most effective features were found to be variance and range of the horizontal representation, and interquartile range and spectral coefficient at 2Hz of the vertical representation. The output of the tree-ensemble is transformed into probability distribution using the scoring provided by the AdaBoost classifier. The probability distribution is then used as an

input for the generative classifier, in our case a Discrete Hidden Markov Model (DHMM), which effectively acts as an state-transition filter. The precision of the kinematic motion classifier exceeds 99% enabling robust separation between high-kinematic pedestrian activities and the rest of the considered transportation modalities.

Stationary Classifier

When the kinematic motion classifier detects a low-kinematic motion environment, the process is passed to a *stationary* classifier, which determines whether the user is stationary or in an automotive transportation. As the stationary classifier, we use an ensemble of $T = 15$ decision trees of depth two. The most expressive features for this task are the peak-features corresponding to peak volume and duration, segment-features corresponding to frequency of acceleration and braking peaks and variance of peak duration over the segment. Among the frame-based features, variance, root mean square, peak spectral coefficient and wavelet entropy were selected from the horizontal representation, and interquartile range, variance, root mean square and spectral coefficient at 2Hz were selected from the vertical representation. The precision of the resulting classifier exceeds 95%, which combined with high performance of the root-level kinematic motion classifier enables identification of automotive transportation with high confidence.

Automotive Classifier

When automotive transportation is detected, the classification proceeds to an *automotive* classifier which is responsible for fine-grained distinction of the current automotive mode as one of the five considered modes: bus, train, metro, tram, and car. The classifier is based on an ensemble of $T = 20$ decision trees of depth two. Segment-based features corresponding to frequency of acceleration and braking peaks were found to be the most important features as they enable distinguishing between vehicles that move alongside other traffic (i.e., car, bus and tram) from vehicles moving independently of other traffic (i.e., train and metro). From peak-based features, maximum intensity and duration of the acceleration and braking periods were selected as effective features in separating car from other motorized modalities, as the quicker driving maneuvers make car distinguishable from larger vehicles. Peak features corresponding to peak intensity and volume of acceleration and braking periods, on the other hand, are effective in separating a slowly moving tram from other motorized modalities. Among

the frame-based features, variance, range, root mean square, entropy, peak spectral coefficient of both horizontal and vertical representations were selected. Additionally, spectral coefficients corresponding to 2, 3 and 5Hz were selected from the vertical representation. These features help in quantifying the overall motion intensity, as well as motion components within key frequencies, and can help to distinguish vehicles operating on roads from those operating on rails.

3.4.3 Performance evaluation

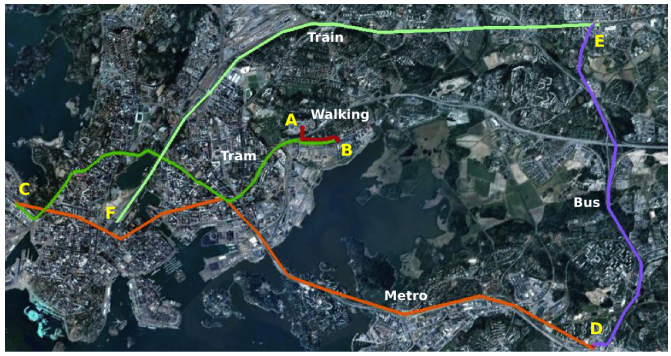
Within this section, we present selected evaluations of our accelerometer-based transportation mode detection, demonstrating the effectiveness of the presented sensing pipeline and the proposed peak- and segment-based features. As the basis of our evaluations, we compare our system with two well-established baselines, i.e., an accelerometer-based system proposed by Wang et al. [143], and a hybrid GPS-accelerometer system proposed by Reddy et al. [116]. Further evaluations covering generalization, power-consumption and robustness of the system, are described in Article I.

Datasets

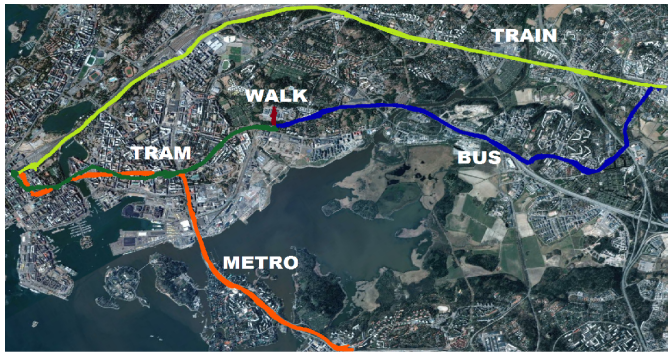
Our full dataset consists of over 150 hours of transportation data collected by 16 volunteers over a period of two years within four different countries. The dataset covers seven different transportation modalities, i.e., stationary, walking, and five automotive modalities: bus, train, tram, metro and car. Here, we focus on a subset of 67 hours collected in supervised settings from two predefined transportation scenarios:

- Scenario A: Collected during winter of $2011-2012$ by nine volunteers, with a total of approximately 44 hours of data.
- Scenario B: Collected during autumn of 2012 by seven volunteers, with a total of approximately 23 hours of data.

The scenarios, illustrated in Figure 3.9a and Figure 3.9b, were designed to include public transportation modalities available in downtown Helsinki, Finland. Between each public transportation modality a short walk was added to collect data also from pedestrian activity. The scenarios lasted approximately from 90 to 120 minutes, varying based on traffic conditions and public transit timetables. The first scenario covers transportation during winter $2011-2012$, allowing collection of data in winter conditions. The second scenario took place in the autumn of 2012 when roads were generally



(a) Scenario A



(b) Scenario B

Figure 3.9: Overview of the scenarios which were used for data collection.

free of ice and snow. Volunteers participating in the two scenarios were selected to minimize overlap between the two scenarios, resulting in 14 unique participants over the two scenarios. To simultaneously cover multiple device placements, the scenario data was collected from three common mobile device placements in an urban space [56]: trouser pockets, bag and jacket pockets. To avoid disturbing the devices collecting sensor measurements, ground truth labels were annotated on an additional mobile phone that was not part of those collecting sensor data.

As the first evaluation, we consider cross-user evaluation using combined data $d = [u_1, u_2, ...u_{14}]$ from the two scenarios, where u_i denotes data corresponding to user i. Specifically, we iterate over data corresponding to individual users $u_{1...14}$, where for each user i, we train the classifier $f_i^*(x;\theta) = \hat{y}$ with training data $d_i^{train} = d - u_i$, i.e., withholding data from user u_i from the training data. The classifiers are then tested with

	Precision			Recall		
TMode	Peaks	Wang	Reddy	Peaks	Wang	Reddy
Stationary	96.1 (0.5)	57.3 (4.5)	81.6 (1.0)	70.0 (2.1)	59.5 (2.3)	70.6 (2.9)
Walk	93.1 (0.1)	87.2 (0.2)	97.7 (0.1)	95.9 (0.1)	89.1 (0.2)	95.9 (0.1)
Bus	78.2 (4.2)	71.1 (1.4)	67.3 (1.6)	78.0 (3.3)	70.4 (1.4)	86.2 (6.4)
Train	68.2 (5.0)	32.1 (0.8)	7.7 (4.4)	80.1 (4.0)	31.6 (0.7)	55.4 (11.9)
Metro	64.5 (5.9)	54.4 (0.6)	70.1 (8.8)	82.0 (2.6)	51.4 (0.9)	56.6 (3.5)
Tram	84.0 (2.1)	58.1 (0.8)	82.8 (7.5)	86.1 (2.1)	58.2 (0.8)	64.5 (7.0)
Mean	80.1 (2.9)	60.0 (1.4)	68.0 (3.9)	82.1 (2.4)	60.2 (1.1)	71.6 (5.3)

Table 3.3: Detection accuracy and variance for our system, and a comparison against the baseline systems.

data from the left-out user u_i, resulting in 14 subsets of results $\hat{y}_i$, where $i = 1...14$. Table 3.3 summarizes results aggregated over all the users for our method as well as for the considered baselines, detailing mean and variance of precision and recall for each transportation mode. Note that car was not included in the scenarios as it was not practically feasible for us to collect driving measurements from the two scenarios.

Cross-user evaluation

From the results, we can observe that the mean precision and recall of our approach is over 80%, with relatively small variance across users. Compared to the two considered baselines, i.e., the approaches of Reddy et al. [116] and Wang et al. [143], our approach provides over 20% higher precision and recall than the approach of Wang et al. and over 10% higher precision and recall than the approach of Reddy et al. The most notable performance differences can be observed for automotive and stationary periods, demonstrating the effectiveness of the peak- and segment-based features.

Cross-placement evaluation

One of the key challenges in motion sensing systems for activity recognition is the impact of device placement on the physical motion measured by the sensors. To investigate how placement of the device affects detection performance, we have conducted cross-placement evaluations. The cross-placement evaluations are analogous to the cross-user evaluations, with the difference that instead of dividing the data per user, we divide the data per device placement, i.e., trouser pockets, bag and jacket pockets. The

TMode	Precision*	Recall*	Precision**	Recall**
Stationary	95.1 (2.0)	72.4 (2.1)	61.9 (-34.2)	64.0 (-6.0)
Walk	92.7 (2.4)	92.4 (2.0)	93.0 (-0.1)	93.0 (-2.9)
Bus	84.8 (3.6)	79.6 (0.5)	71.6 (-6.6)	71.5 (-6.5)
Train	74.7 (4.0)	80.8 (1.9)	25.1 (-43.1)	54.9 (-25.2)
Metro	69.6 (3.8)	81.3 (1.7)	60.1 (-4.4)	56.0 (-26.0)
Tram	88.4 (2.5)	87.9 (1.6)	69.6 (-14.4)	66.7 (-19.4)
Mean	84.2 (3.0)	82.4 (1.6)	63.6 (-16.5)	67.7 (-14.3)

Table 3.4: Detection performance for (*) cross-placement evaluation, and (**) cross-user evaluation without the peak features.

aggregated results over the three placements are shown in the first two columns of Table 3.4. In the results, we can observe that the detection performance for most transportation modes is higher than in the cross-user case, indicating that our approach is able to provide good detection performance over the considered device placements. This result aligns with the earlier observation that placement of the device during mechanized transportation has a lower impact on the sensor measurements compared to the effects of placement variation during pedestrian activities.

Performance of peak and segment features

The main novelty in our accelerometer-based transportation mode detection system is the introduction of peak- and segment-based features. To quantify the impact of these features, we have repeated the cross-user evaluations using only frame-based features without the peak- and segment-based features. The results enlisted in the last two columns Table 3.4 show a significant decrease on detection performance, in particular for the automotive modalities. These results, combined with the comparison with the baselines, provide strong support that the proposed peak- and segment-based features are able to accurately characterize mechanized movement at a granularity that allows distinction of the transportation modalities considered in our study. Finally, we note that the version of our approach using solely frame based features can still attain performance slightly above those obtained with the method proposed by Wang et al. [143], suggesting that the combination of our gravity-estimation approach and frame-based feature engineering are able to produce more expressive motion information compared to the sensing pipeline proposed by Wang et al. [143].

3.5 Summary of Contributions

In the first part of this thesis we have contributed by proposing novel algorithms for gravity component estimation. We have separately considered gravity component estimation within systems relying solely on accelerometers, and when gyroscope measurements are available. We have demonstrated that our approaches are particularly effective during periods of sustained horizontal linear acceleration, enabling better capture of motion information related to these events. To characterize events of sustained linear acceleration, which within our studies correspond to movement within mechanized transportation, we have presented a set of peak- and segment-based features. To showcase the potential of these features, we have presented an accelerometer-based transportation mode detection system able to distinguish between fine-grained automotive modalities. In addition to the evaluations presented within this chapter, the related articles provide a more in-depth description of their performance and generalization.

3.5.1 Additional Evaluations within Article I

In addition to the evaluations presented within this thesis, Article I contains evaluations about:

- Energy-efficiency of the presented accelerometer-based transportation mode detection system, based on empirical power-models constructed using measurements from Monsoon Power Monitor [57]. The results demonstrate 85mW energy consumption for our system, while the baseline of Reddy et al. consumes 240mW and Wang et al. consumes 50mW. We additionally consider a component-wise breakdown of the energy consumption of our approach, and compare it with the energy consumption of typical smartphone applications.

- Generalization of our approach using a separate evaluation based on 65 hours of data collected from everyday settings within Finland, and 20 hours collected from abroad. The results of this evaluation demonstrate unchanged, or even slightly improved, detection accuracy when using data outside the dataset presented in Section 3.4.3.

- Detection robustness in terms of fragmentation and latency using metrics proposed by Ward et al. [144]. The results of this evaluation demonstrate that our approach shows significantly lower fragmentation compared to baselines, but higher latency compared to the GPS-based baseline from Reddy et al.

3.5.2 Additional Evaluations within Article II

In addition to the evaluations presented within this thesis, Article II contains:

- Separate micro-benchmark evaluation of the three IAGE-modules responsible for gravity component estimation. The results of the micro-benchmarks demonstrate that: (i) the Keypoint detection module is triggered during 97% of stationary periods delimited during automotive transportation. When triggered outside stationary periods, the average stability scores indicate that IAGE is able to find reliable keypoints also outside stationary periods. (ii) Keypoint validation is able to detect 95% of false keypoints, with most of the unfiltered keypoints being the first keypoint of a segment. (iii) Orientation change detection is triggered 87.5% of times when a change of more than 5° is imposed on the device.
- Evaluation of horizontal linear acceleration for estimating the overall route distance, with a baseline obtained from route-based ground truth information. The results align with the findings presented in this thesis, demonstrating $10-21\%$ mean error for our approach, and $35-91\%$ mean error for the considered baselines.

Chapter 4

Motion Sensing for Quantifying Pedestrian Activities

Being able to accurately capture motion related to pedestrian activities is an important enabler for a variety of applications across several research fields. For instance, detailed motion information during pedestrian movement has been leveraged within sport applications [5], medical and rehabilitation domain applications [89], and biometric authentication [59]. In addition to identifying the exact type of pedestrian activity [71], accurately capturing motion information during pedestrian movement can be used to augment positioning techniques [49], to track users' walking pace, and to detect intermittent stationary periods delimited within pedestrian segments.

In this chapter, we focus on motion sensing techniques related to quantifying usage of public spaces through a novel technique, coined *Crowd Replication*. As the main technical contribution, we describe how mobile sensing can be leveraged to generate *enriched trajectories* about pedestrian movement and activities within the target area. The enriched trajectories combine automatically inferred position and motion information with manual user annotations, allowing fine-grained quantification of the target area. The contributions presented in this chapter are based on Article IV and Article V.

4.1 Quantifying usage of Public Spaces

Public spaces serve as an important catalyst in modern cities, promoting economic and social well-being of its citizens. In addition to supporting functional services, such as markets, shops, banks and posts, efficient pub-

lic spaces provide an important counterpart for private life [90]. A successful public space functions as a place for relaxation and leisure, and promotes social interaction between people through active and passive engagement [20, 41]. Indeed, even as the role of traditional town squares and piazzas has diminished in contemporary cities, there is a strong consensus among urban researchers that well-designed public spaces remain an important factor for social and psychological health of communities [41, 90, 91].

A successful public space is seen as one promoting a mixture of different types of activities, where the activity types can be categorized as necessary, optional, and social activities [41]. Necessary activities, such as crossing the area to reach a destination, are mostly uncorrelated with the quality of a public space. Optional activities, such as taking a walk for some fresh air or to enjoy the lively atmosphere of an area, depend on the respective qualities of the space. Social activities such as meetings between people, talking or socializing with other people in passing, or children playing together can be seen as resultant of the other two types, i.e., the amount of social activities positively correlates with the quantity of necessary and optional activities present in the area [41]. Consequently, public spaces can be quantified and assessed by measuring the presence of the three activity types within the area. However, collecting representative data about people using the area is a challenging task. The currently available techniques are often too costly, labor-intensive, provide insufficiently detailed information about usage of space, or are not feasible due to privacy intrusion or due to difficulties in obtaining necessary permissions for data collection.

4.1.1 Visual Tracking

Due to the restrictions and limitations related to automatic data collection systems, direct observation remains one of the dominant approaches for quantifying public spaces [53, 91, 102]. For instance, Mehta [90] uses direct observation and manual annotations to record activities within three commercial streets in Boston, USA. The authors subdivided the target area into 78 block segments, each of which is approximately $15 - 18$ meters in length. Based on a pilot study, the authors note that tracking user movement trajectories requires the full attention of the observers, and thus compromises observing and collecting other information types. Consequently, tracking movement was omitted from the actual study, and only stationary and lingering activities were recorded. The authors additionally state that the approach used can quickly cause observer fatigue, in particular within crowded areas, requiring frequent (i.e., every 15 minutes) resting periods to avoid inaccurate or missed measurements. Other studies within urban

planning and architecture have focused on recording movement patterns of people using direct observation and surveys [53, 102]. In a study comparing different approaches for direct observation, Hill [53] notes that accurately tracking user movement trajectories in practice requires following the person. However, if the person becomes aware of the process, the author notes that the user's natural behavior can be affected by so-called observer bias, where the user's normal behavior is altered due to the feeling of being watched.

An alternative approach for tracking pedestrian movement and activities is to use video cameras for automated visual tracking. The recorded video can then be analyzed using either (1) a manual process, where researcher/s annotate activities based on the video recording, or (2) an automated process through the use of computer vision algorithms [109, 141]. The main limitation of the former approach is that manual decoding of a video stream can be an extremely laborious. For instance, in our studies manually decoding a 30-minute video recording required over 50 hours of work, while only producing an aggregated level of information of people's activities [51]. The latter option, i.e., using a computer vision algorithm imposes several requirements for the video recording regarding video resolution, lighting conditions, field of vision and both static and dynamic occlusion [19, 55]. Additional considerations for video-based surveillance include privacy concerns and acquisition of the required permits to set up the video cameras. A more privacy-sensitive solution for automated pedestrian tracking based on laser range scanners was proposed by Shao et al. [122]. However, this solution does not scale to larger spaces. For instance, Shao et al. required eight scanners to measure a 60×15 meter area.

4.1.2 Pervasive Sensing

In indoor environments, pedestrians can be tracked using high precision indoor localization or infrastructure-based tracking. As an example, Larson [73] used RF-ID measurements to track movements of people within a retail environment. Related commercial tracking solutions such as Quuppa HAIP [112] and Ubisense [134] can provide high resolution indoor tracking. The main constraints with these approaches are the requirement that users carry specific hardware, or have their mobile devices configured so that the required sensors, i.e., Bluetooth or WiFi, are continuously on. Due to their commercial nature, these approaches have a high deployment cost, which can be prohibitive to small research groups. Instead of tracking the position of individual people, Bluetooth and WiFi can be leveraged for *crowd*

sensing, i.e., to estimate overall population density [147, 120]. However, these solutions only provide an estimate of the overall number of people using the space and cannot be used to characterize the spatiotemporal distribution of activities, movements, and other behaviors of the people using the space. Additionally, and similarly to positioning users, the requirement of having these sensors continuously switched on and discoverable on the mobile devices can induce uncontrollable and hard-to-estimate biases in the data sample.

Mobile device-based voluntary location tracking using positioning, e.g., GPS, GSM or CDR, can be used to track pedestrian (as well as mechanized) movement over a large area [71, 115]. Such data can be used to analyze macroscopic patterns of cities to estimate aggregated mobility modes, or regional events [18]. However, within public spaces, the granularity at which they can track activities is very coarse and does not adequately address the concerns of designers and planners of human-scale spaces, who require understanding the distribution of necessary, optional and social activities [41].

4.1.3 Motion Sensing for Pedestrian Tracking

Motion sensing can be leveraged to track detailed information about pedestrian activities. Contrary to sensors used in positioning, motion sensors can accurately capture specific pedestrian activities, e.g., whether the user is walking, running or briefly stationary [83]. Motion sensors can also capture other details of the walking activity, such as exact walking pace [51], and changes in movement trajectory [113]. Such information has been shown to be indicative of user attention [2], which can be used to assess attraction of public space features such as info screens or stands.

In addition to tracking details of pedestrian activities, physical motion sensing can assist in tracking user's location through *Pedestrian Dead Reckoning* (PDR). PDR systems track relative location of the user, i.e., the change in position over time. Instead of estimating velocity directly using integration of linear acceleration, a common heuristic in PDR systems is to employ *step detection*, and to calculate displacement by multiplying each step with an estimate of step length. Such systems leveraging step detection to estimate movement on a horizontal plane have been coined *Step and Heading Systems* (SHS) [49]. As the name implies, the two main tasks of an SHS are detection of steps, and heading estimation. Besides these tasks, the distance covered by each step has to be estimated to track changes in velocity and location of the user.

Step Detection

Step detection is one of the classical motion sensing problems with a wealth of studies and techniques available [15, 49]. Traditional approaches for step detection rely on monitoring a suitable motion representation, such as vertical acceleration or the L^2-norm, and apply a threshold-based heuristic to identify steps [47]. A closely related technique is to apply peak detection [78] with a possible addition of a predefined threshold for minimum peak value to identify steps. These approaches are particularly effective when the placement of the sensing device is predefined, and placed in such a body position (e.g., at the foot) that is able to accurately measure acceleration caused by steps. These approaches, however, are sensitive to selection of the minimum peak value or threshold value since motion intensity can vary, e.g., depending on the user's walking style and surface [40]. When operating under unconstrained device use, the challenges related to selection of threshold values are greatly increased due to problems related to varying placement and sensing hardware, as described in Chapter 2.

Another popular approach is to leverage the cyclical pattern associated with human walking [113], which can be used to both identify walking activity, and to detect individual steps. Depending on device placement, the detected cyclical pattern can relate to either individual steps, or to a *stride* corresponding to two steps. The cyclical pattern is typically discovered using auto-correlation [113], or by inspecting key frequency components of the signal spectra [8]. These approaches, however, also require setting threshold values for correlation coefficients or signal amplitudes, and thus are also sensitive to selection of threshold values, sensor placement and changes in it. Additionally, due to relying on consecutive signal patterns with similar characteristics, these approaches are ill suited for step detection in situations where the walking pace is highly varied, or when there are frequent stops delimited within the walk. As a more rarely used technique, steps can also be detected using template matching [155]. Template matching relies on detecting steps by matching motion information signal with a predefined template, typically using either cross-correlation or Dynamic Time Warping (DTW).

To summarize, the main challenges in step detection on mobile devices relate to (i) unconstrained device placement, (ii) producing false positives during non-walk activities, and (iii) step counting during asymmetrical walking, including single steps, shuffling and side-steps. Additionally, there is a need for standard evaluation datasets and metrics, as the reported systems tend to overestimate their performance, usually due to using insufficiently varied evaluation setups [49].

Step Length Estimation

PDR systems generally compute travelled distance by multiplying the detected steps with an estimate of the step length. A simple approach providing a decent distance estimate is to assume a constant step length. A more accurate estimation of the user's step length can be calculated as a function of his/her weight and height. Alternatively, personal step length can be learned during a calibration phase. For instance, the AutoGait [21] system opportunistically learns a linear function between step length and step frequency. To learn the function, the system leverages GPS-augmented data collected outdoors when the user walks in a relatively unconstrained path. Similar approaches learning the relationship between step length and frequency in indoor environments have been proposed by Li et al. [78] and Wang et al. [138]. In both systems, information about a set of predefined reference points and their distances is used to adjust step length estimates.

Heading Estimation

User heading is typically inferred using a combination of inertial sensors and magnetometer. As discussed in Section 2.4.2, inertial sensors can be used to transform measurements from the device's local coordinates to global coordinates respective to inclination Θ. Magnetometer can then be used to derive a compass heading for the device with respect to Earth's magnetic field, allowing fully transforming sensor measurements onto EIC frame. In practice, however, there are several technical problems related to accurate estimation of device orientation respective to Earth's magnetic field, and the actual movement direction of the user.

The first technical challenge relates to the reference frame transformation respective to device inclination. During pedestrian activities with no significant sustained linear acceleration, standard techniques combining accelerometer and gyroscope are typically able to provide a good estimation of the device inclination. The process, however, is not perfect and any errors in inclination transformation directly propagate on the estimation of horizontal magnetic field.

Within estimating the direction of the magnetic north, a more problematic source of errors are produced by electromagnetic disturbances (e.g., hard and soft-iron errors) caused by ferromagnetic metal frames and conducting wires ubiquitously present within modern indoor environments. These dynamic and local disturbances have been shown to produce 15 to 100-degree errors in heading estimation [1]. These errors can, to an extent, be mitigated using a fusion between gyroscope and magnetometer, where

gyroscope is used to detect and filter out large magnetometer errors [63], or correlations between gyroscope and magnetometer are used to determine periods where the magnetometer measurements are sufficiently reliable for heading estimation [138].

Finally, on mobile devices, heading direction of the user is rarely aligned with the compass reading of the device. Consequently, changes in heading direction can initially only be interpreted in relative terms and, to resolve the absolute heading direction, an additional step to estimate the offset between device compass reading and the user's heading direction is required. A common heuristic is to estimate direction of motion from accelerometer measurements using PCA either directly on accelerometer measurements [67], or on horizontal linear acceleration [126]. Alternatively, placement of the phone can be automatically detected using, e.g., audio-cues from a microphone [93], or statistical features from an accelerometer [66]. Placement-dependent models can then be used to estimate the offset between user heading and the compass reading of the device.

4.2 Crowd Replication

Key challenges in data collection methods for quantifying usage of public spaces are to collect data from a representative sample of people using the public space, and to collect data at high precision and resolution. To ensure feasibility of the approach, the data collection method should consider cost and effort of the data collection, address privacy concerns, and be considerate of the required permissions to perform the study.

To address limitations of existing methods for quantifying usage of public spaces, we have developed a sensor-assisted data collection method coined the *Crowd Replication*. Crowd replication has been designed as an efficient, low-cost approach that can be adopted by small research groups for collecting a representative data set from a target area. The underlying idea in crowd replication is to instrument researchers with sensing and annotation tools while they investigate a target area. Specifically, in crowd replication, the researchers are tasked with replicating trajectories and activities of people using the space. The mobile devices carried by researchers then capture a sensor trace of the replicated trajectories, closely mimicking a trace that the target person would have produced if (s)he had been wearing the sensing device(s). Simultaneously to replicating the target's trajectory, the researcher employs a second device to collect annotations about visual observations. To mitigate observer fatigue associated with visual observation, the second device contains a helper application allow-

ing quick insertion of key user demographics, selected activities, and other relevant context to the sensor trace. The annotations are automatically associated with sensor information about location and motion information, thus allowing easy analysis.

The main advantage of crowd replication is the ability to combine benefits from automatic sensing systems to those associated with direct observation. This combination helps crowd replication to capture and distinguish between the three activity categories needed to characterize and assess public spaces, i.e., necessary, optional and social activities [41]. Instrumenting the researcher with sensing devices, compared to deploying the sensing application for the general population using the area, results in important benefits related to the accuracy of the sensing. These benefits include ability to (i) select a sampling design that provides an unbiased sample of activities in the area, and (ii) optimize sensor selection and placement to mitigate the key challenges in activity recognition and PDR systems. For a more detailed comparison of crowd replication and its key benefits compared to other methods, see the attached Article V.

4.3 Pedestrian Tracking using Motion Models

Our approach for pedestrian tracking is based on recording *enriched trajectories* about people traversing through the target area. Enriched trajectories combine details about spatiotemporal movement with motion information detailing pace, intensity and physical activities. The quality of the enriched trajectories directly depend on the context inference solution used to process the location and motion sensor measurements. In crowd replication, we utilize a novel high-accuracy sensing solution for offline motion and location inference. The main advantage of our approach is that it is able to utilize the entire history of collected data for each researcher to learn a personalized model of his/her movement characteristics. Compared to the prevalent research paradigms for motion sensing, our approach inverts the focus of the sensing system, i.e., instead of seeking generalization over an unconstrained and heterogeneous sensing device, we optimize the sensing accuracy when variables such as device placement, model and user specifics can be fixed. In practice, the tailored data inference approach is able to maintain higher accuracy compared to off-the-shelf tracking techniques, especially during varying walking paces.

An overview of our sensing solution is shown in Figure 4.1. The motion sensing pipeline used in our approach includes a preprocessing stage and transformation of measurements into a suitable motion representation. As

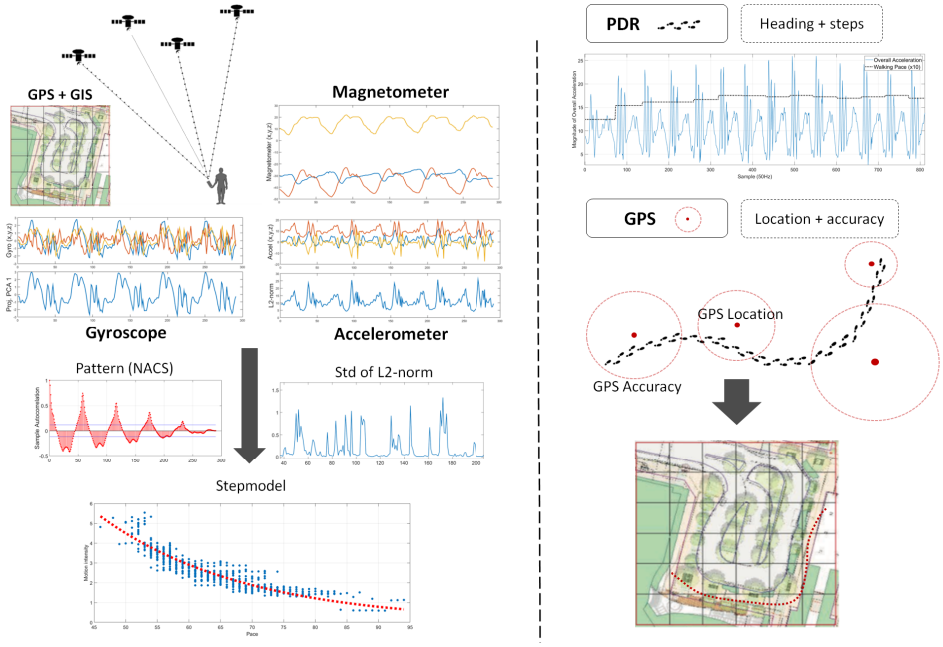


Figure 4.1: Overview of mobile sensing approach.

preprocessing technique, we resample the data to uniform 50Hz frequency and employ a low-pass filter to reduce jitter and noise in the measurements. We then buffer the measurements into 1-second frames, with $k-1$ sample overlap, where k is the number of samples in a data frame. As the main motion representations, we use the L^2-norm of accelerometer measurements to monitor intensity of the motion, and projection along the first principal component of gyroscope measurements to capture the cyclical pattern of the motion.

Identifying activity periods We identify frames with active movement using a threshold on the standard deviation of the overall acceleration [9]. We consider a frame as active whenever the standard deviation exceeds a threshold $\sigma_{active} = 0.30$. This corresponds to a lenient threshold that labels frames with any notable motion as active. We segment the data into active and inactive periods by combining successive frames exceeding the activity threshold σ_{active}. To improve segmentation precision, we identify the exact start and end time of each segments by analyzing the first and last frames, and consider the first sample to violate the threshold σ_{active} as the boundary point of the corresponding segment.

Identifying cyclical movement From the identified activity periods, we next find parameters f_t and f_o for frame length and frame overlap respectively, which optimize the capture cyclical movement in terms of autocorrelation score. For this task, we consider only the motion representation from gyroscope, as we found gyroscope to more robustly capture the cyclical pattern associated with walking. Projecting gyroscope measurements along the first principal component is particularly efficient in our case since we can restrict the placement and orientation of the device due to the instrumented researcher design. We mine the signal for cyclical patterns by adaptively dividing the data into frames using a grid-search that identifies the optimal frame length and overlap parameters to capture the repetitive cycle. Specifically, we vary frame length f_t between $1 - 5$ seconds, and frame overlap f_o between 1 – (k-1) samples to find measurements frames that are optimal in terms of their autocorrelation score. The advantage of our approach is that it can guarantee an optimal framing for the underlying pattern, whereas fixed frame parameters used by most step detection approaches [15] would make the detection susceptible to parameter values that are incompatible with the user's actual walking pattern. Fixed frame parameters would also struggle with identifying walking patterns when the underlying walking pace varies.

Step candidates From the generated data frames, we identify autocorrelation peaks ρ_p that exceed a minimum threshold $\rho_{\min} = 0.30$ as initial candidate step frequencies. Frames where the minimum threshold is exceeded are split into sub-frames, referred to as *step frames*, corresponding to measurements for candidate steps. Autocorrelation lag only provides an estimate of the pace over the entire frame, whereas the duration of each individual step within the frame can vary. To address this variation, we find exact step frame boundaries using minima in overall acceleration as cut-off points. Specifically, we define step frame boundaries by finding local minima from the overall acceleration. To avoid situations where the step duration varies significantly from one step to another, and to account for normal variation in human walking pace, we require these minima to be within 20% of the point corresponding to the initial candidate step frequency. For every detected candidate step, we then store *step features*, i.e., the duration of the step, value of the autocorrelation, minimum, maximum and standard deviation. Regarding the selection of threshold values, we note that the proposed approach to instrument researchers with sensing devices(s) allows controlling many of the most challenging variables in motion sensing systems, e.g., device model, device placement and user

characteristics. This is a key benefit allowing robust selection of threshold values when compared to systems that are designed to work in the wild and have to generalize over such variables.

Clustering step candidates To reduce the computational complexity of the successive processing stages where the step frames are used to learn regression models, we reduce the number of samples through clustering. We cluster candidate step frames based on similarity in pace f, motion intensity m and autocorrelation ρ. This is particularly effective during continuous unobstructed walk, which typically exhibits a stable pace and overall motion, producing multiple steps with high similarity. The clusters are formed by processing through the frames in order of descending autocorrelation score. Each sample is tested against existing clusters and added to the model it best fits. If no model provides a good fit (currently we use $d \geq 0.05$ as threshold) for the sample, we initiate a new cluster with the sample. Formally, the distance between sample A and cluster C is calculated as:

$$d = abs(f_A - f_C)/f_C + abs(\rho_A - \rho_C)/\rho_C + abs(m_A - m_C)/m_C \quad (4.1)$$

Step models Using the clustered step features, we learn a (regression) step model between pace and motion intensity, i.e., between duration of candidate steps and standard deviation of overall acceleration during the candidate step. From the collected data, we observe that the relation between walking pace and motion intensity is roughly linear for normal walking pace, and in case of very slow or fast walking, the relation roughly follows an exponential decay function curve. Typically there are one to three such relationally dependent subsets within the data, each corresponding to a pattern generated by a step multiple. In practice, depending on walking pace and device placement, the strongest pattern is produced by either a repetition of a single step or repetition of two steps, often referred to as a *stride*. The pattern for a single step is typically stronger when the device is symmetrically affected by steps taken by both legs, e.g., while held in hand. Stride, on the other hand, produces stronger pattern when the device is asymmetrically affected by steps, e.g., when the device is placed within the user's trouser pocket.

Based on these observation, we learn models for the first two orders as we found these to be the most relevant, while the third and consecutive orders are redundant to the first two. This process follows a two-stage approach. First, we initialize two models with seed samples corresponding to the clusters formed around the frame with the strongest autocorrelation.

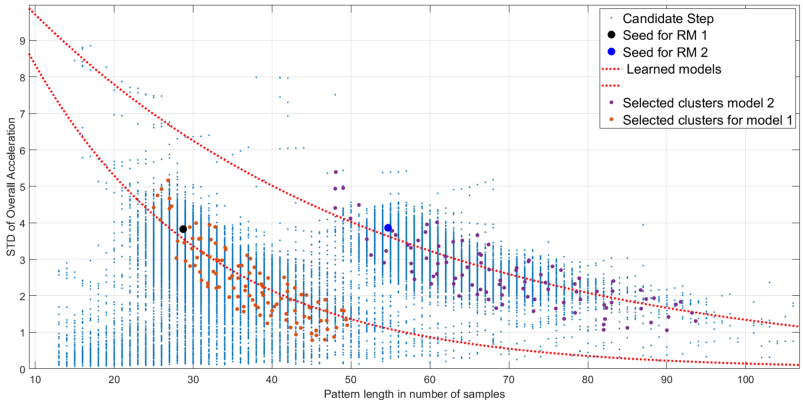


Figure 4.2: Regression models discovered from the data between walking pace and standard deviation of accelerometer $L2$-norm. In the model $N_1 = 100700$, $RMSE_1 = 0.22$, $R-Squared_1 = 0.8750$, and N_2=203910, $RMSE_2 = 0.31$, $R - Squared_2 = 0.6536$

We then find feasible candidate clusters that exhibit a downward sloping linear relation between walking pace and motion intensity. As there is natural variability in both step length and the motion intensity values, we allow a degree of slack (currently 5% of the model's initial sample) to fulfill the requirement for negative linear relation. We note that at this stage clusters can be valid for more than one model. We then follow an iterative process to find a set of clusters which produce the most cohesive regression models. During each iteration we relearn a regression model, and in case the fit is below minimum requirement, remove the cluster with the highest residual from the model. To account for the varying number of samples forming the clusters, we use the cluster size as a weight when learning the model. The process is continued until a sufficiently good fit of the data is obtained, or the number of remaining data points falls below a threshold. If some clusters remain in both models after the filtering process, we assign each of these conflicting clusters to the model where it produces the smaller residual. The final step models learned from data are illustrated in Figure 4.2.

Once the step models have been learned, candidate steps are validated against the model to identify those that exhibit a good fit. We note that there is also an inherited autocorrelation threshold $\rho > .30$ as the candidates are generated only for frames above the threshold. The final outcome in terms of pace aligned over raw motion intensity data is shown in Figure 4.3.

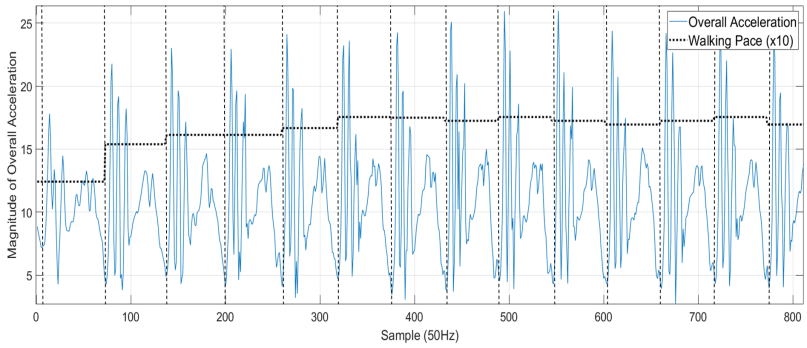


Figure 4.3: Illustration of detected walking pace (in seconds, scaled by 10) aligned with raw data about overall acceleration.

4.3.1 Integration with Location Sensing

We recreate movement trajectories of the replicated subjects by combining location information obtained from a suitable positioning technology, and relative change in location obtained using pedestrian dead reckoning (PDR). In our experiments, we focus on an outdoor space, and hence we rely on GPS as our underlying positioning system. For experiments focusing on indoor spaces, leveraging an existing indoor positioning technique could reasonably replace GPS; see Article V for discussion. We sample GPS at 1Hz, which ensures high tracking accuracy. Note that, while GPS suffers from heavy battery drain, the sensing application can run continuously for several hours on a contemporary smartphone, which is sufficient for typical data collection in crowd replication. For extended data collection tasks, replacement phones might need to be considered to avoid draining the phone's battery. Our PDR approach is based on detecting steps and pace using the pedestrian modeling module as described in the previous Section 4.3, and a heading estimation from sensor fusion-based orientation filtering method proposed by Madgwick [84]. As an additional source of location information, in our use-case we leverage a sampling design (i.e., method to select participants for the study) that takes advantage of the natural entry points leading in and out of the target area. Each replication is thus started from one of these entry points, providing an accurate starting location that can be used to initialize the spatiotemporal movement trajectory.

We fuse the different information sources to produce final spatiotemporal movement trajectories following established best practices for hybrid GPS and PDR location tracking. Specifically, GPS measurements are used

to manage accumulation of error from PDR, whereas PDR is used to improve location accuracy between the GPS measurements, in particular when GPS reception is poor. We balance between GPS measurements and PDR estimates by weighting GPS location estimates with the horizontal accuracy reported by GPS. Specifically, we allocate the location measurement a weight $\alpha = [0.0, 0.5]$ using a decay function $f(x)$ so that $f(x \geq 15.0m) = 0.0$ and $f(x \leq 5.0m) = 0.5$. Exponential decay is used to determine the appropriate weight for the intermediate values between $x = [5, 15]$ meters.

4.3.2 Accuracy of Pedestrian Modeling

We next assess the accuracy of our solution for pedestrian modeling and demonstrate that it outperforms current state-of-art techniques. We carry out the evaluation using a benchmark dataset collected by Brajdic and Harle [15]. The dataset contains accelerometer measurements from walking behavior which have been collected from 117 trials with 21 users. Ground truth is provided for 102 of these traces. The dataset covers 7 typical device placements and each trial features a period of fast, normal and slow walking. As the data only contains accelerometer measurements, we modify our pedestrian modeling by substituting gyroscope with accelerometer measurements smoothed with a centered moving average filter with a 10ms window length. For other parts our pedestrian modeling remains unchanged and follows the steps described in Section 4.3. We separately consider step recognition accuracy with and without walk detection. In the former case we consider traces that have been labeled as walking, and in the latter case we consider the full benchmark dataset. To provide a baseline for evaluation, we implemented the best performing techniques from the paper of Brajdic and Harle [15]. For step counting we implemented a windowed step detection algorithm, and for walk detection we used a threshold on overall acceleration (L2-norm). The best performance for the windowed step detection was obtained with parameter values $\text{MovAvr}_{win} = 0.28$s and $\text{PeakWin}_{win} = 0.4$s.

Results for the walk segments are shown in Figure 4.4. The median error rate of our technique is 1.7% across all walking periods, with the highest errors resulting when the phone is placed on the upper body or in a backpack. When the device is held in hand or placed in a trouser pocket, which are the most typical placements in crowd replication, the errors are consistently below 1%. In contrast, the median error of the baseline is 2.3%. The baseline performs slightly better for handbag and backpack placements where the signals are generally symmetric, but has a high error

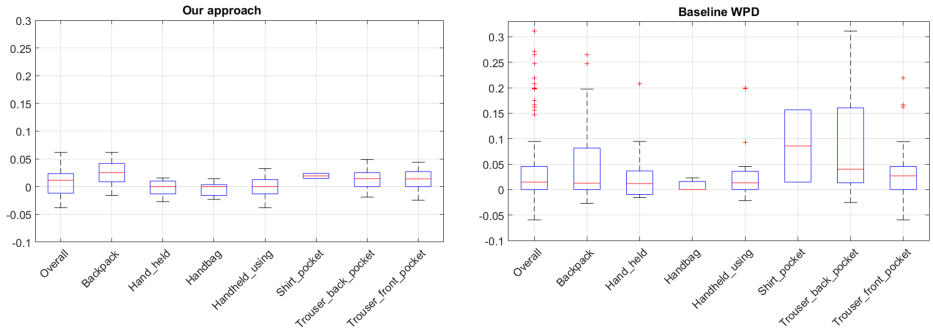


Figure 4.4: Boxplots for accuracy of step evaluation in the first case, where we only evaluate over known walking periods. The left-hand figure provides results per device placement with our approach, and the right-hand figure provides results for the baseline with windowed peak detection.

rate for trouser pocket where the signal is likely to be asymmetric. As shown by Muaaz and Mayrhofer [97], mimicked walking behavior often contains asymmetry, implying that our technique is likely to perform even better for replicated data than the baseline. Current state-of-the-art step recognition approaches remain highly effective for *online* detection. However, in crowd replication all measurements can be analyzed offline, which enables our approach to improve on step recognition performance.

The results across all segments, i.e., when step recognition needs to be combined with walking detection, are shown in Figure 4.5. The error rate of our technique remains low, with handbag placement being the sole exception. The baseline performance, however, is significantly impacted by a high number of false positives, achieving only 21.5% accuracy, which would affect the total step count. Our approach is only slightly impacted by additional data, obtaining a median overall accuracy of 4.1%, demonstrating the benefits and need for our pedestrian modeling approach. Indeed, the error rate of the state-of-the-art algorithms would significantly overestimate the amount of walking, and thus result in inaccurate indicators and activity estimates.

4.4 Summary of Contributions

In the second part of the thesis, we have contributed by presenting the concept of instrumenting researchers with sensing devices, and thus inverting the generalization problem by focusing on optimizing sensing accuracy

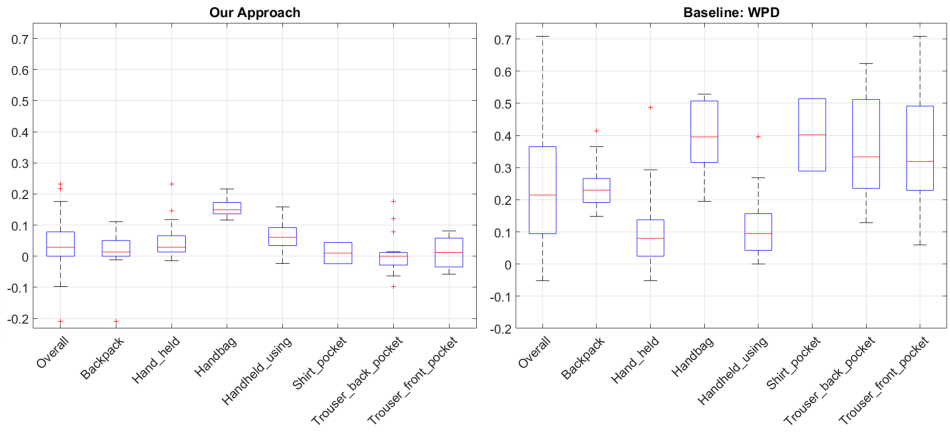


Figure 4.5: Box plots for accuracy of step evaluation in the second case, where we evaluate the entire trace and rely on walking detection to identify walking periods. The left-hand figure provides results per device placement with our approach, and right-hand figure provides results for the baseline with windowed peak detection.

when a predetermined configuration of sensing devices can be fixed. As the main technical contribution, we have presented construction of pedestrian models, which are able to learn a personalized model for a given user-placement combination and thus increase accuracy of step and pace detection when compared to state-of-art approaches. Based on these concepts, we have presented crowd replication as a novel, sensor-assisted approach for collecting enriched trajectories about users' movement and activities within a target space.

In addition to the evaluations presented in this thesis, the related Article IV and Article V present a use-case of crowd replication in assessing the quality of a typical metropolitan public space. The main findings from this study are summarized as:

- We evaluate the validity of crowd replication as a data collection mechanism by demonstrating that the data collected with crowd replication converges to the characteristics of the target population. The results of this evaluation demonstrate that after approximately $50 - 250$ user replications demographics such as gender, age and group size converge. The movement trajectories converge in approximately 300 replications, indicating a point where the majority of movement trajectories and their variants have been observed.

- We evaluate the accuracy of the spatiotemporal trajectories by comparing aggregated dwell time between crowd replication and ground truth obtained from a video recording. Aggregated dwell time refers to the sum of all people passing through the area, split into 15x15m cells. As the result of this evaluation, we find correlation 0.93 between crowd replication and the baseline, which is further increased to 0.98 when smoothed over neighboring cells.

- We critically compare data collection effort of crowd replication and other data collection mechanisms, demonstrating that crowd replication is able to combine benefits from more resource and labor intensive methods, while requiring only commodity devices and a few active volunteers.

- We demonstrate that crowd replication is able to capture fine-grained quantification of activities, liveliness [90], and movement within the space by presenting results derived from the example use-case.

Chapter 5

Applications for Motion Sensing

In this chapter we outline two exemplary applications of motion sensing within mobile applications. The first application, *Matkahupi*, described in Article III, is a persuasive mobility application promoting sustainable transportation behavior through the use of transportation mode detection. The second application, *CeeSR* described in Article VI, is a collaborative speech monitoring application that leverages motion sensing to detect changes in physical configuration of the collaborating devices.

5.1 Matkahupi: Persuasive Mobile Application for Sustainable Mobility

As an example of the real-world applications of a transportation mode detection system, we have integrated a variation of the accelerometer-based transportation mode detection system in a persuasive transportation application called *Matkahupi* [60]. The underlying concept of the Matkahupi application is similar to the one proposed by Froehlich et al. in UbiGreen [38], i.e., to automatically track users' transportation behavior to estimate transportation-related CO_2-emissions, and use persuasive elements to promote more sustainable transportation behavior.

Within Matkahupi, depicted in Figure 5.1, we have integrated five core functionalities: (i) a journey planner considering the public transportation available in the target area, (ii) an automatic transportation mode detection module, (iii) a travel diary with a log of past journeys, (iv) tracking and visualizing CO_2-emissions related to transportation, and (v) a persuasive component based on actionable *challenges* personalized for the user and the detected transportation behavior. Specifically, after each detected

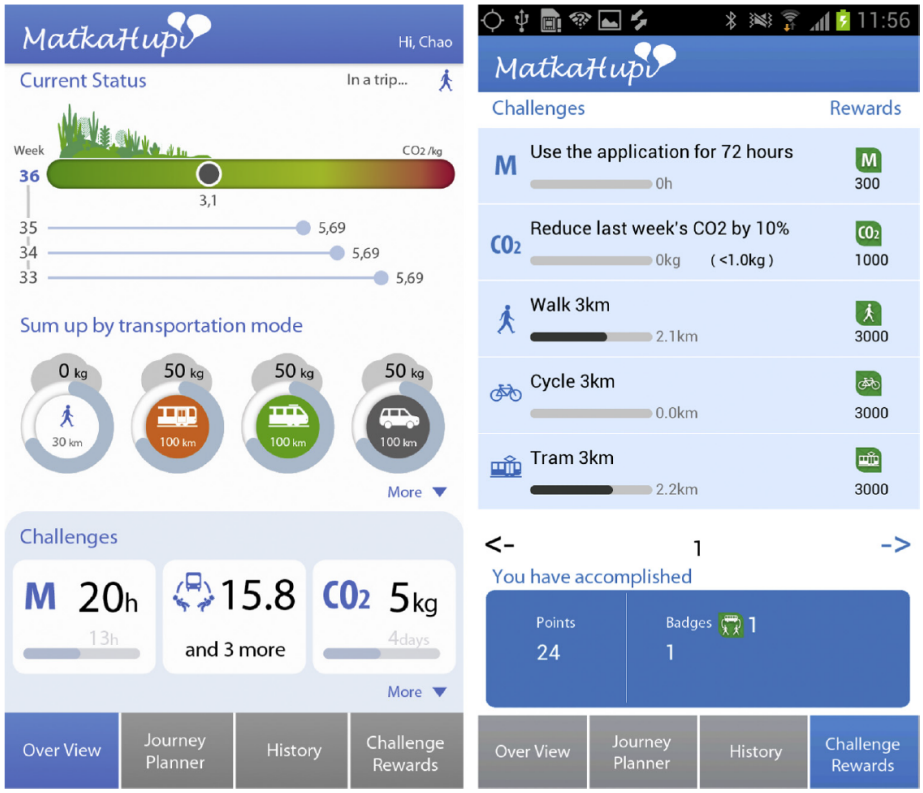


Figure 5.1: Screen capture of the two main Activity screens of MatkaHupi. On the left screen, the user is shown his/her weekly CO_2-emissions and its distribution by transportation mode (top and middle), the current active challenges (bottom) and the current transportation mode (top right). On the right screen the user is shown a summary of the ongoing challenges (top and middle), total number of points attained (bottom right), and badges earned (bottom left).

trip, the application explores alternative transportation methods to check if the same trip could have been made faster and/or with less emissions. The alternative travel plan options are personalized to consider users' transportation preferences and limitations. When feasible options are found, the application prompts a *trip challenge* to the user to complete the same trip using the alternative travel plan. In addition to trip challenges based on detected transportation, the application provides *general challenges* prompting the user to, e.g., walk or cycle progressively longer journeys, or try out a new transportation modality such as tram or metro. Finally, the application contains progressive *weekly challenges* prompting a reduction of transportation-related weekly emissions by 10%. After completing a challenge, the application grants the user a specific amount of points or a badge, providing both an incentive to complete the proposed challenges, and a positive feedback once a challenge is completed [26].

The transportation mode detection module operating at the core of Matkahupi is responsible for automatic identification between nine modalities: stationary, walking, running, cycling, bus, train, tram, metro, and car. The transportation mode detection system in Matkahupi augments motion information with location measurements obtained from GPS and WiFi sensors. The decision to leverage location information in the transportation mode detection system was largely due to the application already tracking user locations for other functionalities, rendering the additional cost of location sensing minimal. Additionally, the transportation mode detection system was later required to identify the exact public transit lines and operators, requiring mapping user location with an external GIS database detailing the underlying transportation network.

No formal evaluation on the performance of the transportation mode detection module was performed. Instead, based on a pilot study involving 12 users over approximately 4 weeks, with weekly questionnaires and an interview conducted at the end of the study, the transportation mode detection was seen as performing overall well. Additionally, the challenges based on automatic transportation mode detection received positive feedback, with 70% of the participants considered the application useful for monitoring transportation-related emissions, 70% stating that the application had motivated them to walk more during the experiment, and 40% of the participants considering changing their travel behavior. However, the received feedback also pointed out that the battery-consumption and heating of the phone were seen as significant limitations to continued use of the application. While data communication was responsible for a large part of the overall energy-consumption, the continuous GPS location tracking was

deemed as the second component causing unfeasible high energy consumption. The pilot study thus provided tangible evidence on the potential of automated transportation mode detection within the scope of persuasive mobility applications, while also providing evidence on the importance of energy-efficiency within such applications.

5.2 CeeSR: Collaborative and Energy-efficient Speech Monitoring on Smart Devices

Besides specialized motion sensing domains, the ability to detect and quantify movement can act as an important support functionality within systems that are sensitive to physical movement. As an example application, we have implemented a motion detection module within a system called *CeeSR*, a collaborative and energy-efficient speech monitoring system.

CeeSR enables capturing high-quality speech in everyday settings, supporting innovative applications such as collaborative work applications [133] and life-logging applications [75]. Contrary to existing solutions which require either instrumenting either the user [151] or the environment [107] with specific hardware, CeeSR opportunistically leverages commodity devices such as smartphones, smart watches, or other programmable devices equipped with microphones present in a given situation. The main benefits provided by CeeSR are (i) reduced overall energy-cost associated with producing the audio recording, as the burden for recording is distributed between participating devices, and (ii) increased audio quality through unification of the individual audio streams into a composite stream, as the individual audio streams can complement each other. The collaborative sensing process consists of three main submodules, illustrated in Figure 5.2. In the following sections, we describe the aim and overall operating logic of each submodule, after which we describe the motion detection module within CeeSR. For a more thorough description of each module, we refer to Article VI.

Intelligent Audio Source Selection Devices co-located within close proximity of each other are likely to capture similar audio streams. While these audio streams can potentially complement each-other, there simultaneously is significant redundancy in the captured audio. Consequently, the overall energy cost associated with producing the audio record can be significantly reduced by switching off a subset of the present devices. To determine the optimal set of devices for the recording, CeeSR incorporates

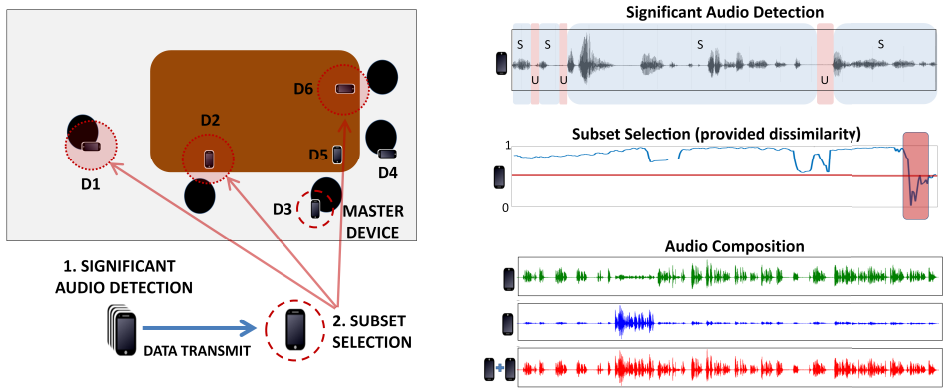


Figure 5.2: Overview of the CeeSR sensing process given the scenario depicted on the left. As the first step, significant audio events are detected and relayed to the device acting as the group master, here D3 in the left-hand picture and the top graph on the right-hand picture. Next, during intelligent audio source selection, the subset of devices providing optimal balance between energy-efficiency and quality of the recording is resolved, corresponding here in the left-hand picture to devices D1, D2, D6 and in the right-hand picture to the middle graph. Finally, selected audio sources are fused into a composite audio stream, here depicted in the bottom graph of the right-hand picture. © IEEE 2016.

an intelligent collaborator selection algorithm, which is based on comparing the total dissimilarity of audio (i.e., potential added information) captured on a device with the cost of audio sampling, and by identifying a subset that together maximizes overall internal dissimilarity while minimizing cost of sampling.

Significant Audio Event Detection Depending on the placement relative to the active speaker, microphone occlusion and noise, devices participating in the speech monitoring commonly capture only fractions of the actual audio. The periods when there is no significant audio present thus provide opportunities to reduce the overall energy cost associated with the speech monitoring. CeeSR leverages this notion by incorporating a lightweight significant audio activity detection algorithm that is able to adapt to the prevalent audio environment and operate without a training phase. The algorithm is based on comparing an autocorrelation-derived measure with an adaptive background noise threshold to determine whether the frame contains significant audio activity, and only shares frames with significant audio with the master device.

Audio Alignment and Composition In the final stage, the individual audio streams are unified into a single composite stream. To align the signals, we find and utilize the time lag that maximizes correlation between two signals. The volume of different signals are normalized by weighting each signal with a coefficient corresponding to a dynamic gain multiplier, calculated as the combined Root Mean Square of the individual signals forming the composition.

Motion Detection The selected subset of devices only remains valid as long as the physical configuration of the device remains relatively stable. Consequently, CeeSR leverages motion information to detect changes in physical device configuration. Specifically, we use a two-stage motion detection module that detects potentially significant changes in device placement. First, variation in accelerometer magnitude are used to identify events of significant physical activity. Second, events of significant physical activity are filtered using a Hidden Markov Model (HMM) to reduce false positives. When significant changes in physical configuration of the devices are observed, recording is reinitialized on all devices and the optimal group composition is recalculated.

Chapter 6

Discussion and Conclusion

In this thesis, we have investigated motion sensing techniques and their use within selected representative application domains. We have proposed new methods for both physical motion sensing and machine learning-based motion inference to capture detailed motion information during transportation, and to quantify movement within specific areas. In the follow sections, we outline limitations we have identified within our methods, and propose directions for future research.

6.1 Limitations

While the proposed systems and techniques generally perform well in the considered use cases, there are some limitations to consider. Some of these limitations pave the way for potential future research, whereas other can be considered to be engineering challenges in real-world implementations. In the following we outline some of the main limitations within the proposed techniques.

6.1.1 Gravity Estimation

The proposed methods for gravity estimation presented in Section 3.1 and Section 3.2 are inherently heuristic, performing well in typical situations but are vulnerable to breakage when the underlying assumptions do not hold. For instance, sustained linear acceleration characterized by long and gradual acceleration during a period with low overall motion and noise can be confused with opportune periods for gravity estimation. Such situations can result in generation of false keypoints which are hard to detect and filter, resulting in loss of motion information about the sustained linear acceleration period. Reliably preserving motion information about linear

acceleration in such situations might require careful inspection of the overall magnitude of acceleration, which in turn would require highly accurate calibration of the sensors. In our work, we have avoided the requirement of high-quality sensors, nor do we require virtual recalibration of the sensors. This has been an intentional design principle to retain maximal generality of our approach. For applications particularly sensitive to errors in gravity estimation, more constrained requirements about sensor quality and calibration might be required. Given well-calibrated sensors, the proposed gravity estimation methods could be extended to take advantage of the additional information. More careful investigation of such extensions is here left as future research direction.

Within IMU-based adaptive gravity estimation (IAGE), we note that the approach used in the orientation detection module to detect and validate orientation changes is susceptible to producing latency in gravity estimation in case of sustained high-variance movement. This is because the orientation change event does not complete as long as high-variance motion persists. This limitation could be alleviated by progressively validating orientation change instead of waiting for the event to complete. Within IAGE, linear acceleration is decomposed to tangential and radial components based on correlation between rotation around gravity component Ω_G^z, and linear acceleration on global horizontal plane $\acute{A}_G^{x,y}$. We note an alternative method for determining the tangential and radial components of horizontal linear acceleration is to first perform principal component analysis (PCA) on the horizontal linear acceleration. During mechanized transportation, this procedure typically yields tangential and radial components, which can then be identified using correlation with rotation around the vertical global plane.

6.1.2 Peak Features and Accelerometer-based Transposition mode detection

Within the presented accelerometer-based transportation mode detection, data collection scenarios have focused on ensuring a non-ambiguous and precise ground truth. For instance, careful attention was paid to remaining still while within automotive modalities to avoid mixed activities within the ground truth data. This results in data sets which can be too ideal compared to normal behavior, which more frequently involves moving within automotive transportation, frequent interactions with the device, and other sources of external activities. This problem is alleviated, to an extent, by evaluations with everyday data collection. However, the data collectors have been aware of the process and the restrictions used in the data col-

lection scenarios, and might have followed similar behavior when collecting data during everyday transportation.

While the peak and segment features show encouraging performance for identification of automotive modalities, we acknowledge an inherent risk within transportation mode detection of capturing specific routes instead of transportation modes. For instance, a specific train service might exhibit almost invariable patterns of acceleration and braking periods due to traversing the same line, resulting in segment features which can accurately capture the train service. However, the main contributor for the segment features in such a scenario is the route instead of the transportation mode. This limitation can be alleviated by ensuring a sufficiently varied set of training data.

As a practical concern, continuous monitoring of motion sensor(s) is often impractical within the major operating systems of the mobile devices, which force various limits to performing continuous background sensing and processing. In specific operating systems and versions, e.g., certain Android OS versions, motion sensors can still be sampled continuously by using internal buffering and motion co-processors. While allowing the collection of full motion sensor tracers, the buffering process prohibits real-time motion inference techniques. In such situations, the proposed methods might be better suited for delayed or offline interpretation of the collected sensor data, diminishing the importance of real-time performance.

6.1.3 Pedestrian models for Crowd Replication

The data collection technique we have employed in Crowd Replication requires following people within the target area. We acknowledge that the replication process should be carried out in a way that avoids any privacy or ethical concerns, and socially awkward situations to either the target user or the researcher performing the replication. The necessary constraints depend on the characteristics of the target space, but should be planned so that replication can be conducted without the target user becoming aware or disturbed by the replication. This is also crucial for reducing possible observer biases in the collected sample. In some cases, tracking a participant might need to be aborted to avoid revealing the replication process. Generally, when only few people are using the space and/or they are mostly stationary, it is better to use an alternative technique, such as paper annotations or video recording combined with post-hoc replication, to capture the data. People staying within the area during the replication process, e.g., due to working, might also become aware of the process by noticing the same people going repeatedly back-and-forth in the area. While not

directly problematic for data collection, we acknowledge that this might cause confusion and even observer bias to the stationary activities. Potential ways to mitigate these issues are to inform the people working at the target area of the process, split the replication into a series of shorter trials, or periodically swap the researchers responsible for replicating users.

Crowd replication is mainly designed to capture information about movement trajectories within the target space. Stationary and lingering activities are less suited for replication, as prolonged shadowing could result in the replicated people becoming aware of the process. Replication of stationary activities additionally ties up one researcher for the period of the activity, leading to non-optimal use of human resources. Instead of replicating stationary or lingering activities, such activities are generally better captured using complementing technique, such as the direct observation methods proposed by Mehta [91].

For large spaces, e.g., ones spanning several streets, crowd replication may be unsuitable. Firstly, shadowing another person for a prolonged period of time is prone to making the user aware of the replication process. Additionally, the researcher carrying out the replication may start to lose concentration, which could result in inaccurate data (e.g., decrease quality of mimicking or result in erroneous or missing labels). Secondly, while errors in our location and pedestrian tracking are very small, they accumulate over time and hence data quality would decrease. A potential solution for such spaces would be to divide the area into sub-regions and to perform crowd replication within these sub-regions.

6.2 Future Work

The contributions related to the gravity component and linear acceleration estimation presented in this thesis have direct implications for a multitude of applications relying on device inclination estimation, e.g., adapting orientation of device display, tilt correction for magnetometers, and device placement detection. Interesting future research directions emerge from investigating the severity of errors produced by inaccurate gravity component estimation during periods of sustained linear accelerations within these applications, and quantifying what the attainable benefits are from correct gravity component estimation.

When employed for mechanized transportation, pursuing further usage of the improved horizontal motion representation can pave the way for a wealth of future studies. For instance, exploring to what extent location trajectories can be recreated while using only mobile device-embedded mo-

tion sensors would have impact on privacy and security domains, while also potentially enhancing existing hybrid inertial-location sensing solutions for tracking a user's location. Combined with peak-features, the more accurate estimation of linear acceleration can benefit studies related to assessing vehicular maneuvers, which could then be used as enablers for applications such as driving skill and behavior assessment, monitoring traffic conditions, and to detect dangerous traffic incidents. So far, the presented research has mainly focused on horizontal linear acceleration. As a future research direction, the vertical component could also be more carefully considered to support applications more reliant on movement within vertical and horizontal components, such as aerial or underwater vehicles. Additionally, linear acceleration associated with land-based mechanized transportation frequently occurs partly on vertical dimension, e.g., on slopes and hills, banked turns or elevated motorway junctions.

Within transportation mode detection, the ability to identify transportation mode while only relying on accelerometer acts as an enabler of privacy-sensitive and low-energy solutions for tracking users' transportation behavior. Such solutions could be applied to collection of automated travel diaries, usable for urban and transportation planners. Besides smartphones, such an approach would be suitable for other mobile and wearable devices including only a three-dimensional accelerometer. We envision, for instance, that travel cards could be equipped with a low-power accelerometer, enabling tracking users transportation behavior automatically. Another interesting future research direction is opened by the increased temporal scale of information obtained from each transportation trip. As the scale of information expands from frame-based information to containing entire routes, an important research question arises whether an approach is capturing the characteristics of the *route*, the transportation mode or even the user. While undesirable from the perspective of transportation mode detection, being able to identify certain routes or a route type could be interesting for transportation sciences, for instance to identify a specific route without requiring information about its location trajectory.

As an on-going research we are currently pursuing, we are in the process of exploring generalization of the presented accelerometer-based transportation mode detection, in particular within the light of the limitations detailed above in Section 6.1. Additionally, to support the more wide-spread availability of gyroscopes, we are in the process of working on IMU-based transportation mode detection work. Inclusion of gyroscope enables us to leverage IAGE as part of the overall system, allowing more accurate gravity component estimation and thus also more robust motion representations

about mechanized movement. As a part of these studies, we are seeking to compare the methods with state-of-art deep learning approaches, as well as to take advantage of the standard datasets made available by recent advances on the field [43].

6.3 Conclusion

We have presented novel algorithms for estimation of the gravity component for both accelerometer-only-based systems (AAGE), and for IMU-based systems (IAGE). The main advantage of the proposed systems is the ability to more accurately estimate the gravity component during periods of sustained linear acceleration. As an example application demonstrating effectiveness of the proposed techniques, we have applied AAGE for accelerometer-based transportation mode detection system. The more accurate capture of sustained linear acceleration during mechanized movement is leveraged by introducing novel peak- and segment-based features, which are able to quantify key aspects of vehicular movement. We have then detailed a classification framework able to distinguish between fine-grained automotive modalities, demonstrating significant improvements in detection accuracy compared to previous work while using solely accelerometer.

Within crowd replication, we have presented a novel high-accuracy pedestrian motion tracking system. Our approach is based on an assumption of instrumented researchers, taking advantage of the fixed device model, placement, and orientation of the sensing devices. In our experiments, we demonstrate that the presented motion sensing approach is able to outperform existing off-the-shelf techniques for walking, step and pace detection. Within crowd replication, we demonstrate that the proposed approach is able to accurately capture indicators about human behavior within the target space, while simultaneously providing a cost-effective, privacy-sensitive and unbiased data collection method.

Finally, we have presented two exemplary applications that leverage motion information to execute their key functionalities. In the first example, Matkahupi, motion information is used to implement the core module responsible for identification of current the transportation mode. In the second example application, CeeSR, motion information is used as a support module ensuring that the application's main functionality for collaborative speech monitoring is not ill-affected by the physical motion of the participating devices.

References

[1] M. Afzal, V. Renaudin, and G. Lachapelle. Assessment of indoor magnetic field anomalies using multiple magnetometers. In *23rd International Technical Meeting of the Satellite Division of the Institute of Navigation*, 2010.

[2] F. Alt, A. Bulling, L. Mecke, and D. Buschek. Attention, please!: Comparing features for measuring audience attention towards pervasive displays. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS 16*. ACM Press, 2016.

[3] Apple. App store. `https://www.apple.com/lae/ios/app-store/`.

[4] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275 – 286, 2003.

[5] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *23th International Conference on Architecture of Computing Systems 2010*, pages 1–10, Feb. 2010.

[6] M. Bachlin, M. Plotnik, D. Roggen, I. Maidan, J. Hausdorff, N. Giladi, and G. Troster. Wearable assistant for parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):436–446, mar 2010.

[7] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Proceedings of the 2nd International Conference on Pervasive Computing (PERVASIVE)*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer-Verlag, 2004.

[8] P. Barralon, N. Vuillerme, and N. Noury. Walk detection with a kinematic sensor: Frequency and wavelet comparison. In *2006 Interna-*

tional Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, aug 2006.

[9] S. Bhattacharya, H. Blunck, M. Kjærgaard, and P. Nurmi. Robust and energy-efficient trajectory tracking for mobile devices. *IEEE Transactions on Mobile Computing*, 14:2, 2015.

[10] S. Bhattacharya and N. D. Lane. From smart to deep: Robust activity recognition on smartwatches using deep learning. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, mar 2016.

[11] R. Bhoraskar, N. Vankadhara, B. Raman, and P. Kulkarni. Wolverine: Traffic and road condition estimation using smartphone sensors. In *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*. IEEE, jan 2012.

[12] J. R. Blum, D. G. Greencorn, and J. R. Cooperstock. Smartphone Sensor Reliability for Augmented Reality Applications. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (MobiQuitous)*, 2012.

[13] A. Bolbol, T. Cheng, I. Tsapakis, and J. Haworth. Inferring hybrid transportation modes from sparse gps data using a moving window svm classification. *Computers, Environment and Urban Systems*, 31;6:526–537, 2012.

[14] A. Bourke, J. O'Brien, and G. Lyons. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & Posture*, 26(2):194–199, jul 2007.

[15] A. Brajdic and R. Harle. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 225–234. ACM, 2013.

[16] A. Bulling, U. Blanke, and B. Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, 46:33: 1 – 33, 2014.

[17] J. B. J. Bussmann, W. L. J. Martens, J. H. M. Tulen, F. C. Schasfoort, H. J. G. van den Berg-Emons, and H. J. Stam. Measuring daily behavior using ambulatory accelerometry: The activity monitor. *Behavior Research Methods, Instruments, & Computers*, 33(3):349–356, aug 2001.

[18] F. Calabrese, L. Ferrari, and V. D. Blondel. Urban sensing using mobile phone network data: a survey of research. *ACM Computing Surveys (CSUR)*, 47(2):25, 2015.

[19] J. Candamo, M. Shreve, D. B. Goldgof, D. B. Sapper, and R. Kasturi. Understanding transit scenes: A survey on human behavior-recognition algorithms. *IEEE transactions on intelligent transportation systems*, 11(1):206–224, 2010.

[20] S. Carr, M. Francis, L. Rivlin, and A. Stone. *Public Space, Environment and Behavior Series*. Cambridge University Press, 1992.

[21] D.-K. Cho, M. Mun, U. Lee, W. J. Kaiser, and M. Gerla. AutoGait: A mobile platform that accurately estimates the distance walked. In *Proceedings of the 8th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 116–124. IEEE Computer Society, 2010.

[22] F. Chollet and H. Liu. *A (not so) short introduction to MEMS*. memscyclopedia.org, 2018.

[23] T. Choudhury, G. Borriello, S. Consolvo, D. Haehnel, B. Harrison, B. Hemingway, J. Hightower, P. P. Klasnja, K. Koscher, A. LaMarca, J. A. Landay, L. LeGrand, J. Lester, A. Rahimi, A. Rea, and D. Wyatt. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, 7(2):32–41, 2008.

[24] I. Cleland, B. Kikhia, C. Nugent, A. Boytsov, J. Hallberg, K. Synnes, S. McClean, and D. Finlay. Optimal placement of accelerometers for the detection of everyday activities. *Sensors*, 13(7):9183–9200, jul 2013.

[25] S. Colton. The balance filter: a simple solution for integrating accelerometer and gyroscope measurements for a balancing platform. White paper, Massachusetts Institute of Technology, 2007.

[26] S. Consolvo, P. Klasnja, D. W. McDonald, D. Avrahami, J. Froehlich, L. LeGrand, R. Libby, K. Mosher, and J. A. Landay. Flowers or a robot army?: encouraging awareness & activity with personal, mobile displays. In *Proceedings of the 10th international conference on Ubiquitous computing (Ubicomp)*, pages 54–63. ACM, 2008.

[27] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith,

and J. A. Landay. Activity sensing in the wild: a field trial of ubifit garden. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1797–1806, New York, NY, USA, 2008. ACM.

[28] J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, jan 2007.

[29] R. Curey, M. Ash, L. Thielman, and C. Barker. Proposed IEEE inertial systems terminology standard and other inertial sensor standards. In *PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No.04CH37556)*. IEEE, 2004.

[30] S. Dabiri and K. Heaslip. Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation Research Part C: Emerging Technologies*, 86:360–371, jan 2018.

[31] J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16):1–35, 2006.

[32] B. T. Division. Lexington area travel data collection test. *Global Positioning Systems for Personal Travel Surveys*, 1997.

[33] Y. Endo, H. Toda, K. Nishida, and A. Kawanobe. Deep feature extraction from trajectories for transportation mode estimation. In *Advances in Knowledge Discovery and Data Mining*, pages 54–66. Springer International Publishing, 2016.

[34] J. Farringdon, A. Moore, N. Tilbury, J. Church, and P. Biemond. Wearable sensor badge and sensor jacket for context awareness. In *Proceedings of the 3rd International Symposium on Wearable Computers (ISWC)*, pages 107 –113. IEEE, 1999.

[35] T. Feng and H. J. P. Timmermans. Transportation mode recognition using GPS and accelerometer data. *Transportation Research Part C*, 37:118 – 130, 2013.

[36] D. Figo, P. Diniz, D. Ferreira, and J. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14-7:645–662, 2010.

[37] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of*

the Second European Conference on Computational Learning Theory, 1995.

[38] J. Froehlich, T. Dillahunt, P. Klasnja, J. Mankoff, S. Consolvo, B. Harrison, and J. A. Landay. Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits. In *Proceedings of the 27th international conference on Human factors in computing systems (CHI)*, pages 1043–1052. ACM, 2009.

[39] D. Gafurov, K. Helkala, and T. Søndrol. Biometric gait authentication using accelerometer sensor. *Journal of Computers*, 1(7), nov 2006.

[40] D. Gafurov and E. Snekkenes. Towards understanding the uniqueness of gait biometric. In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*. IEEE, sep 2008.

[41] J. Gehl. *Life between buildings: using public space*. Island Press, 2011.

[42] E. Gilman, A. Keskinarkaus, S. Tamminen, S. Pirttikangas, J. Röning, and J. Riekki. Personalised assistance for fuel-efficient driving. *Transportation Research Part C: Emerging Technologies*, 58:681–705, Sept. 2015.

[43] H. Gjoreski, M. Ciliberto, L. Wang, F. J. O. Morales, S. Mekki, S. Valentin, and D. Roggen. The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices. *IEEE Access*, 6:42592–42604, 2018.

[44] M. Glueck, D. Oshinubi, and Y. Manoli. Automatic realtime offset calibration of gyroscopes. In *2013 IEEE Sensors Applications Symposium Proceedings*. IEEE, feb 2013.

[45] Google. Google play. `https://play.google.com/store`.

[46] U. Goverment. Selective availability. `https://www.gps.gov/systems/gps/modernization/sa/`. [Retrieved April 2019].

[47] P. Goyal, V. J. Ribeiro, H. Saran, and A. Kumar. Strap-down pedestrian dead-reckoning system. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–7. IEEE, 2011.

[48] H. Han, M.-J. Kim, and J. Kim. Development of real-time motion artifact reduction algorithm for a wearable photoplethysmography. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, aug 2007.

[49] R. Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys & Tutorials*, 15(3):1281–1293, 2013.

[50] E. A. Heinz, K. S. Kunze, M. Gruber, D. Bannach, and P. Lukowicz. Using wearable sensors for real-time recognition tasks in games of martial arts - an initial experiment. In *2006 IEEE Symposium on Computational Intelligence and Games*. IEEE, may 2006.

[51] S. Hemminki, K. Kuribayashi, K. Shin'ichi, P. Nurmi, and S. Tarkoma. Crowd replication: Sensing-assisted quantification of human behaviour in public spaces. To appear in: ACM Transactions on Spatial Algorithms and Systems (TSAS).

[52] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2013.

[53] M. Hill. Stalking the urban pedestrian: A comparison of questionnaire and tracking methodologies for behavioral mapping in large-scale environmnets. *Environment and Behavior*, 16(5):539, 1984.

[54] K. Hinckley, J. Pierce, M. Sinclair, and E. Horvitz. Sensing techniques for mobile interaction. In *Proceedings of the 13th annual ACM symposium on User interface software and technology - UIST '00*. ACM Press, 2000.

[55] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3):334–352, 2004.

[56] F. Ichikawa, J. Chipchase, and R. Grignani. Where's the phone? a study of mobile phone location in public spaces. In *Proceedings of the 2nd International Conference on Mobile Technology, Applications and Systems*, pages 1 – 8. IEEE, 2005.

[57] M. S. Inc. Monsoon power monitor. `https://www.msoon.com/`. [Retrieved April 2019].

[58] D. A. Johnson and M. M. Trivedi. Driving style recognition using a smartphone as a sensor platform. In *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITS)*, 2011.

[59] F. Juefei-Xu, C. Bhagavatula, A. Jaech, U. Prasad, and M. Savvides. Gait-ID on the move: Pace independent human identification using cell phone accelerometer dynamics. In *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, sep 2012.

[60] A. Jylhä, P. Nurmi, M. Siren, S. Hemminki, and G. Jacucci. Matkahupi: a persuasive mobile application for sustainable mobility. *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 227–230, 2013.

[61] M. Kauppila, S. Pirttikangas, X. Su, and J. Riekki. Accelerometer based gestural control of browser applications. 05 2008.

[62] N. Kern, B. Schiele, and A. Schmidt. Multi-sensor activity context detection for wearable computing. In E. H. L. Aarts, R. Collier, E. van Loenen, and B. E. R. de Ruyter, editors, *Proceedings of the 1st European Symposium on Ambient Intelligence (EUSAI)*, volume 2875, pages 220–232. Springer, 2003.

[63] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park. A step, stride and heading determination for the pedestrian navigation system. *Journal of Global Positioning Systems*, 3:273–279, Dec. 2004.

[64] M. Kok, J. D. Hol, and T. B. Schön. Using inertial sensors for position and orientation estimation. *Foundations and Trends® in Signal Processing*, 11(1-2):1–153, 2017.

[65] A. Krause, M. Ihmig, E. Rankin, D. Leong, S. Gupta, D. Siewiorek, A. Smailagic, M. Deisher, and U. Sengupta. Trading off prediction accuracy and power consumption for context-aware wearable computing. In *Proceedings of the 9th IEEE International Symposium on Wearable Computers (ISWC)*, volume 0, pages 20–26, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[66] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster. Where am i: Recognizing on-body positions of wearable sensors. In *Proceedings of the 1st International Workshop on Location- and Context-Awareness (LoCA)*, pages 264–275. Springer, 2005.

[67] K. S. Kunze, P. Lukowicz, K. Partridge, and B. Begole. Which way am i facing: Inferring horizontal device orientation from an accelerometer signal. In *Proceedings of the 13th IEEE International Symposium on Wearable Computers (ISWC)*, pages 149–150, 2009.

[68] H. Kwon, G. D. Abowd, and T. Plötz. Adding structural characteristics to distribution-based accelerometer representations for activity recognition using wearables. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers - ISWC 18*. ACM Press, 2018.

[69] W. L. The history of accelerometer 1920s-1996 - prologue and epilogue, 2006. *Sounds & Vibration*, 41:84–92, Jan. 2007.

[70] C. Ladha, N. Y. Hammerla, P. Olivier, and T. Plötz. Climbax: skill assessment for climbing enthusiasts. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp13*. ACM Press, 2013.

[71] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9):140 –150, 2010.

[72] O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013.

[73] J. S. Larson, E. T. Bradlow, and P. S. Fader. An exploratory look at supermarket shopping paths. *International Journal of Research in Marketing*, 22:395–414, 2005.

[74] S. L. Lau, I. Köning, K. David, B. Parandian, C. Carius-Dussel, and M. Schultz. Supporting patient monitoring using activity recognition with a smartphone. *7th International Symposium on Wireless Communication Systems (ISWCS*, pages 810–814, 2010.

[75] M. L. Lee and A. K. Dey. Lifelogging Memory Appliance for People with Episodic Memory Impairment. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, 2008.

[76] J. Leppänen, M. Pelkonen, H. Guo, S. Hemminki, P. Nurmi, and S. Tarkoma. Collaborative and energy-efficient speech monitoring on smart devices. *Computer*, 49(12):22–30, dec 2016.

[77] J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. In *Proceedings of the 4th International Conference on Pervasive Computing (PERVASIVE)*, volume 3968 of *Lecture Notes in Computer Science LNCS*, pages 1 – 16. Springer - Verlag, 2006.

[78] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of The 2012 ACM Conference on Ubiquitous Computing (UbiComp)*, pages 421–430. ACM, 2012.

[79] Q. Li. Noise reduction of accelerometer signal with singular value decomposition and savitzky-golay filter. *Journal of Information and Computational Science*, 10(15):4783–4793, oct 2013.

[80] L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171:311–331, 2007.

[81] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery - DMKD 03*. ACM Press, 2003.

[82] J. C. Lötters, J. Schipper, P. H. Veltink, W. Olthuis, and P. Bergveld. Procedure for in-use calibration of triaxial accelerometers in medical applications. *Sensors and actuators A: Physical*, 68:221–228, 1998.

[83] H. Lu, J. Yang, Z. Liu, N. D. Lane, C. T., and C. A. The Jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 71–84, 2010.

[84] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *Proceedings of the IEEE International Conference on Rehabilitation Robotics*, pages 1–7. IEEE, 2011.

[85] R. Mahony, T. Hamel, and J.-M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, jun 2008.

[86] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, jul 1989.

[87] D. Marjoux, D. Baumgartner, C. Deck, and R. Willinger. Head injury prediction capability of the HIC, HIP, SIMon and ULP criteria. *Accident Analysis & Prevention*, 40:1135 – 1148, 2008.

[88] Markus Wüstenberg and Henrik Blunck and Kaj Grønbæk and Mikkel Baun Kjærgaard. Distinguishing Electric Vehicles from Fossil-Fueled Vehicles with Mobile Sensing. In *Proceedings of the IEEE 15th International Conference on Mobile Data Management (MDM)*, 2014.

[89] S. Mazilu, M. Hardegger, Z. Zhu, D. Roggen, G. Troester, M. Plotnik, and J. Hausdorff. Online detection of freezing of gait with smartphones and machine learning techniques. In *Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcare*. IEEE, 2012.

[90] V. Mehta. Lively streets determining environmental characteristics to support social behavior. *Journal of planning education and research*, 27(2):165–187, 2007.

[91] V. Mehta. Evaluating public space. *Journal of Urban Design*, 19(1):53 – 88, 2014.

[92] G. Meijer, K. Westerterp, F. Verhoeven, H. Koper, and F. ten Hoor. Methods to assess physical activity with special reference to motion sensors and accelerometers. *IEEE Transactions on Biomedical Engineering*, 38(3):221–229, mar 1991.

[93] E. Miluzzo, M. Papandrea, N. D. Lane, and H. Lu. Pocket, bag, hand, etc. - automatically detecting phone context through discovery. In *Proc. of the First International Workshop on Sensing for App Phones (PhoneSense)*, 2010.

[94] D. Minnen, T. Starner, I. Essa, and C. Isbell. Discovering characteristic actions from on-body sensor data. In *10th IEEE International Symposium on Wearable Computers*, pages 11 – 18. IEEE, 2006.

[95] D. Mizell. Using gravity to estimate accelerometer orientation. In *Proc. Seventh IEEE International Symposium on Wearable Computers*, pages 252–253, 18–21 Oct. 2005.

[96] P. Mohan, V. N. Padmanabhan, and R. Ramachandran. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys)*, pages 323–336. ACM, 2008.

[97] M. Muaaz and R. Mayrhofer. Smartphone-based gait recognition: From authentication to imitation. *IEEE Transactions on Mobile Computing*, 2017.

[98] S. Nawaz and C. Mascolo. Mining users' significant driving routes with low-power sensors. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys)*, 2014.

[99] U. N. D. of Economic and S. Affairs. World population prospects, the 2017 revision, key findings and advance tables. online, 2017.

[100] F. Ordóñez and D. Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, jan 2016.

[101] A. Oulasvirta, T. Rattenbury, L. Ma, and E. Raita. Habits make smartphone use more pervasive. *Personal and Ubiquitous Computing*, 16(1):105–114, jun 2011.

[102] H. Park and S. Satoh. Analysis on the pedestrian rambling activities and spatial structure in downtown. *Journal of Architecture and Planning (Transactions of AIJ)*, 605:143–150, 2006.

[103] V. M. N. Passaro, A. Cuccovillo, L. Vaiani, M. D. Carlo, and C. E. Campanella. Gyroscope technology and applications: A review in the industrial perspective. *Sensors*, 17(10):2284, oct 2017.

[104] D. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In *Proceedings of the 5th International Conference on Ubiquitous Computing (UBICOMP)*, 2003.

[105] L. Paul. *An Introduction to Optimal Estimation.* Addison-Wesley Pub. Co., 1967.

[106] M. Perttunen, O. Mazhelis, F. Cong, M. Kauppila, T. Leppänen, J. Kantola, J. Collin, S. Pirttikangas, J. Haverinen, T. Ristaniemi, and J. Riekki. Distributed road surface condition monitoring using mobile phones. In *Ubiquitous Intelligence and Computing*, pages 64–78. Springer Berlin Heidelberg, 2011.

[107] F. Pianesi, N. Mana, A. Cappelletti, B. Lepri, and M. Zancanaro. Multimodal recognition of personality traits in social interactions. In *Proc. of the 10th international conference on Multimodal interfaces (ICMI)*, 2008.

[108] S. Pirttikangas, K. Fujinami, and T. Nakajima. Feature selection and activity recognition from wearable sensors. In *Proceedings of the Third International Symposium on Ubiquitous Computing Systems (UCS 2006)*, volume 4239 of *Lecture Notes in Computer Science*, pages 516 – 527. Springer-Verlag, 2006.

[109] Placemeter. `https://www.placemeter.com/`, 2019 (accessed January 9, 2019).

[110] T. Plötz, N. Y. Hammerla, and P. Olivier. Feature learning for activity recognition in ubiquitous computing. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[111] A. C. Prelipcean, G. Gidófalvi, and Y. O. Susilo. Transportation mode detection – an in-depth review of applicability and reliability. *Transport Reviews*, 37(4):442–464, oct 2016.

[112] Quuppa. `http://quuppa.com/`, 2019 (accessed January 9, 2019).

[113] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of The 18th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 293–304. ACM, 2012.

[114] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In M. M. Veloso and S. Kambhampati, editors, *Proceedings of the 17th Innovative Applications of Artificial Intelligence Conference (IAAI)*, pages 1541–1546. AAAI Press, 2005.

[115] J. Reades, F. Calabrese, A. Sevtsuk, and C. Ratti. Cellular census: Explorations in urban data collection. *IEEE Pervasive Computing*, 6(3):30–38, 2007.

[116] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks*, 6(2):13:1–13:27, 2010.

[117] RunScribe. Runscribe. `https://runscribe.com/`. [Retrieved April 2019].

[118] K. Sankaran, M. Zhu, X. Guo, A. L. Ananda, M. C. Chan, and L. Peh. Using mobile phone barometer for low-power transportation context detection. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys)*, 2014.

[119] G. Santos. Road transport and CO 2 emissions: What are the challenges? *Transport Policy*, 59:71–74, oct 2017.

[120] L. Schauer, M. Werner, and P. Marcus. Estimating crowd densities and pedestrian flows using wi-fi and bluetooth. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, 2014.

[121] Y. Schutz and A. Chambaz. Could a satellite-based navigation system (GPS) be used to assess the physical activity of individuals on earth? *European Journal of Clinical Nutrition*, 51(5):338–339, may 1997.

[122] X. Shao, H. Zhao, K. Nakamura, K. Katabira, R. Shibasaki, and Y. Nakagawa. Detection and tracking of multiple pedestrians by using laser range scanners. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[123] D. Shin, D. Aliaga, B. Tunçer, S. M. Arisona, S. Kim, D. Zünd, and G. Schmitt. Urban sensing: Using smartphones for transportation mode classification. *Computers, Environment and Urban Systems*, 53:76–86, sep 2015.

[124] T. Sohn, A. Varshavsky, A. LaMarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold, and E. de Lara. Mobility detection using everyday GSM traces. In *Proceedings of the 8th International Conference on Ubiquitous Computing (Ubicomp)*, pages 212–224, 2006.

[125] X. Song, H. Kanasugi, and R. Shibasaki. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2618–2624, 2016.

[126] U. Steinhoff and B. Schiele. Dead reckoning from the pocket - an experimental study. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 162 – 170. IEEE, 2010.

[127] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu. Transportation mode detection using mobile phones and gis information. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 54–63, 2011.

[128] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems - SenSys 15*. ACM Press, 2015.

[129] P. Stopher, Q. Jiang, and C. Fitzgerald. Processing gps data from travel surveys. *28th Australasian Transport Research Forum, ATRF 05*, 2005.

[130] P. R. Stopher, P. Bullock, and F. Horst. Exploring the use of passive GPS devices to measure travel. In *Applications of Advanced Technologies in Transportation (2002)*. American Society of Civil Engineers, jul 2002.

[131] Thomas Stockx and Brent Hecht and Johannes Schöning. SubwayPS: towards smartphone positioning in underground public transportation systems. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2014.

[132] P. J. Troped, M. S. Oliveira, C. E. Matthews, E. K. Cromley, S. J. Melly, and B. A. Craig. Prediction of Activity Mode with Global Positioning System and Accelerometer Data. *Medicine & Science in Sports & Exercise*, 40:972 – 978, 2008.

[133] S. Tucker, O. Bergman, A. Ramamoorthy, and S. Whittaker. Catchup: A Useful Application of Time-travel in Meetings. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (CSCW)*, 2010.

[134] Ubisense. `http://ubisense.net/en`, 2019 (accessed January 9, 2019).

[135] P. Veltink, H. Bussmann, W. de Vries, W. Martens, and R. V. Lummel. Detection of static and dynamic activities using uniaxial accelerometers. *IEEE Transactions on Rehabilitation Engineering*, 4(4):375–385, 1996.

[136] R. W. Devaul and S. Dunn. Real-time motion classi cation for wearable computing applications. Technical report, MIT Media Laboratory, 2001.

[137] H. Wang, H. Luo, F. Zhao, Y. Qin, Z. Zhao, and Y. Chen. Detecting transportation modes with low-power-consumption sensors using recurrent neural network. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, oct 2018.

[138] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: unsupervised indoor localization. In *The 10th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 197–210. ACM, 2012.

[139] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, and D. Roggen. Benchmarking the SHL recognition challenge with classical and deep-learning pipelines. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers - UbiComp 18*. ACM Press, 2018.

[140] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, and D. Roggen. Enabling reproducible research in sensor-based transportation mode recognition with the sussex-huawei dataset. *IEEE Access*, 7:10870–10891, 2019.

[141] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015.

[142] L. Wang and D. Roggen. Sound-based transportation mode recognition with smartphones. *IEEE ICASSP 2019: Spatial Audio Recording and Detection and Classification of Acoustic Scenes and Events*, 2019.

[143] S. Wang, C. Chen, and J. Ma. Accelerometer based transportation mode recognition on mobile phones. In *Asia-Pacific Conference on Wearable Computing Systems*, pages 44–46, 2010.

[144] J. A. Ward, P. Lukowicz, and H. W. Gellersen. Performance metrics for activity recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):6:1–6:23, 2011.

[145] P. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, jun 1967.

[146] S. Wells, P. Forbes, J. Masthoff, S. Gabrielli, and A. Jyllha. Superhub: Integrating digital behaviour management into a novel sustainable urban mobility system. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*, BCS-HCI '13, pages 62:1–62:2, 2013.

[147] J. Weppner and P. Lukowicz. Bluetooth based collaborative crowd density estimation with mobile phones. In *Proceedings of the IEEE international conference on Pervasive computing and communications (PerCom),*, 2013.

[148] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt. WreckWatch: Automatic traffic accident detection and notification with smartphones. *Mobile Networks and Applications*, 16(3):285–303, mar 2011.

[149] J. Wolf, R. Guensler, and W. Bachman. Elimination of the travel diary: Experiment to derive trip purpose from global positioning system travel data. *Transportation Research Record: Journal of the Transportation Research Board*, 1768(1):125–134, jan 2001.

[150] O. J. Woodman. An introduction to intertial navigation. Technical Report UCAM-CL-TR-696, University of Cambridge, August 2007.

[151] D. Wyatt, T. Choudhury, J. Bilmes, and J. A. Kitts. Inferring colocation and conversation networks from privacy-sensitive audio with implications for computational social science. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2:7:1–7:41, 2011.

[152] Z. Xiao, Y. Wang, K. Fu, and F. Wu. Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. *ISPRS International Journal of Geo-Information*, 6(2):57, feb 2017.

[153] Z. Xiao, H. Wen, A. Markham, and N. Trigoni. Robust pedestrian dead reckoning (r-PDR) for arbitrary mobile device placement. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, oct 2014.

[154] Y. Ye and P. Nurmi. Gestimator: Shape and stroke similarity based gesture recognition. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction - ICMI 15*. ACM Press, 2015.

[155] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek. Automatic step detection in the accelerometer signal. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, pages 80–85. Springer Berlin Heidelberg.

[156] Y. Zheng, Y. Chen, Q. Li, and W.-Y. Xie, X. Ma. Understanding transportation modes based on gps data for web applications. *ACM Transactions on the Web*, 4,1, 2010.

[157] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 312–321, 2008.

[158] Y. Zheng, X. Xie, and W.-Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engineering Bulletin*, 33:32–40, 2010.

[159] W. zhong Wang, Y. wei Guo, B. yu Huang, G. ru Zhao, B. qiang Liu, and L. Wang. Analysis of filtering methods for 3d acceleration signals in body sensor network. In *International Symposium on Bioelectronics and Bioinformations 2011*. IEEE, nov 2011.

TIETOJENKÄSITTELYTIETEEN OSASTO
PL 68 (Pietari Kalmin katu 5)
00014 Helsingin yliopisto

DEPARTMENT OF COMPUTER SCIENCE
P.O. Box 68 (Pietari Kalmin katu 5)
FI-00014 University of Helsinki, FINLAND

JULKAISUSARJA **A**

SERIES OF PUBLICATIONS **A**

Reports are available on the e-thesis site of the University of Helsinki.

A-2014-1 J. Korhonen: Graph and Hypergraph Decompositions for Exact Algorithms. 62+66 pp. (Ph.D. Thesis)

A-2014-2 J. Paalasmaa: Monitoring Sleep with Force Sensor Measurement. 59+47 pp. (Ph.D. Thesis)

A-2014-3 L. Langohr: Methods for Finding Interesting Nodes in Weighted Graphs. 70+54 pp. (Ph.D. Thesis)

A-2014-4 S. Bhattacharya: Continuous Context Inference on Mobile Platforms. 94+67 pp. (Ph.D. Thesis)

A-2014-5 E. Lagerspetz: Collaborative Mobile Energy Awareness. 60+46 pp. (Ph.D. Thesis)

A-2015-1 L. Wang: Content, Topology and Cooperation in In-network Caching. 190 pp. (Ph.D. Thesis)

A-2015-2 T. Niinimäki: Approximation Strategies for Structure Learning in Bayesian Networks. 64+93 pp. (Ph.D. Thesis)

A-2015-3 D. Kempa: Efficient Construction of Fundamental Data Structures in Large-Scale Text Indexing. 68+88 pp. (Ph.D. Thesis)

A-2015-4 K. Zhao: Understanding Urban Human Mobility for Network Applications. 62+46 pp. (Ph.D. Thesis)

A-2015-5 A. Laaksonen: Algorithms for Melody Search and Transcription. 36+54 pp. (Ph.D. Thesis)

A-2015-6 Y. Ding: Collaborative Traffic Offloading for Mobile Systems. 223 pp. (Ph.D. Thesis)

A-2015-7 F. Fagerholm: Software Developer Experience: Case Studies in Lean-Agile and Open Source Environments. 118+68 pp. (Ph.D. Thesis)

A-2016-1 T. Ahonen: Cover Song Identification using Compression-based Distance Measures. 122+25 pp. (Ph.D. Thesis)

A-2016-2 O. Gross: World Associations as a Language Model for Generative and Creative Tasks. 60+10+54 pp. (Ph.D. Thesis)

A-2016-3 J. Määttä: Model Selection Methods for Linear Regression and Phylogenetic Reconstruction. 44+73 pp. (Ph.D. Thesis)

A-2016-4 J. Toivanen: Methods and Models in Linguistic and Musical Computational Creativity. 56+8+79 pp. (Ph.D. Thesis)

A-2016-5 K. Athukorala: Information Search as Adaptive Interaction. 122 pp. (Ph.D. Thesis)

A-2016-6 J.-K. Kangas: Combinatorial Algorithms with Applications in Learning Graphical Models. 66+90 pp. (Ph.D. Thesis)

A-2017-1 Y. Zou: On Model Selection for Bayesian Networks and Sparse Logistic Regression. 58+61 pp. (Ph.D. Thesis)

A-2017-2 Y.-T. Hsieh: Exploring Hand-Based Haptic Interfaces for Mobile Interaction Design. 79+120 pp. (Ph.D. Thesis)

A-2017-3 D. Valenzuela: Algorithms and Data Structures for Sequence Analysis in the Pan-Genomic Era. 74+78 pp. (Ph.D. Thesis)

A-2017-4 A. Hellas: Retention in Introductory Programming. 68+88 pp. (Ph.D. Thesis)

A-2017-5 M. Du: Natural Language Processing System for Business Intelligence. 78+72 pp. (Ph.D. Thesis)

A-2017-6 A. Kuosmanen: Third-Generation RNA-Sequencing Analysis: Graph Alignment and Transcript Assembly with Long Reads. 64+69 pp. (Ph.D. Thesis)

A-2018-1 M. Nelimarkka: Performative Hybrid Interaction: Understanding Planned Events across Collocated and Mediated Interaction Spheres. 64+82 pp. (Ph.D. Thesis)

A-2018-2 E. Peltonen: Crowdsensed Mobile Data Analytics. 100+91 pp. (Ph.D. Thesis)

A-2018-3 O. Barral: Implicit Interaction with Textual Information using Physiological Signals. 72+145 pp. (Ph.D. Thesis)

A-2018-4 I. Kosunen: Exploring the Dynamics of the Biocybernetic Loop in Physiological Computing. 91+161 pp. (Ph.D. Thesis)

A-2018-5 J. Berg: Solving Optimization Problems via Maximum Satisfiability: Encodings and Re-Encodings. 86+102 pp. (Ph.D. Thesis)

A-2018-6 J. Pyykkö: Online Personalization in Exploratory Search. 101+63 pp. (Ph.D. Thesis)

A-2018-7 L. Pivovarova: Classification and Clustering in Media Monitoring: from Knowledge Engineering to Deep Learning. 78+56 pp. (Ph.D. Thesis)

A-2019-1 K. Salo: Modular Audio Platform for Youth Engagement in a Museum Context. 97+78 pp. (Ph.D. Thesis)

A-2019-2 A. Koski: On the Provisioning of Mission Critical Information Systems based on Public Tenders. 96+79 pp. (Ph.D. Thesis)

A-2019-3 A. Kantosalo: Human-Computer Co-Creativity - Designing, Evaluating and Modelling Computational Collaborators for Poetry Writing. 74+86 pp. (Ph.D. Thesis)

A-2019-4 O. Karkulahti: Understanding Social Media through Large Volume Measurements. 116 pp. (Ph.D. Thesis)

A-2019-5 S. Yaman: Initiating the Transition towards Continuous Experimentation: Empirical Studies with Software Development Teams and Practitioners. 81+90 pp. (Ph.D. Thesis)

A-2019-6 N. Mohan: Edge Computing Platforms and Protocols. 87+69 pp. (Ph.D. Thesis)

A-2019-7 I. Järvinen: Congestion Control and Active Queue Management During Flow Startup. 87+48 pp. (Ph.D. Thesis)

A-2019-8 J. Leinonen: Keystroke Data in Programming Courses. 56+53 pp. (Ph.D. Thesis)

A-2019-9 T. Talvitie: Counting and Sampling Directed Acyclic Graphs for Learning Bayesian Networks. 70+54 pp. (Ph.D. Thesis)

A-2019-10 J. Toivonen: Modeling and Learning Monomeric and Dimeric Transcription Factor Binding Motifs. 61+109 pp. (Ph.D. Thesis)